REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	3.Nov.99	MAJOR REPORT

4. TITLE AND SUBTITLE
SALSA: SECURITY APPLICATION LAUNCHER FOR SYSTEM ADMINISTRATORS

5. FUNDING NUMBERS

6. AUTHOR(S)
2D LT DERRICK DOUGLAS C

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TEXAS A&M UNIVERSITY

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
FY99-392

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

19991117 074

14. SUBJECT TERMS	15. NUMBER OF PAGES
	27
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

SALSA: SECURITY APPLICATION LAUNCHER FOR SYSTEM ADMINISTRATORS

A Master Project

by

DOUGLAS C. DERRICK

Submitted to Dr. Udo W. Pooch
Distributed Security Systems
Texas A&M University

in partial fulfillment of the requirements
for Master of Computer Science Degree

Decemeber 12, 1997

Major Subject: Computer Science

ABSTRACT

SALSA. (December 1997)

Douglas C. Derrick,

B.S. United States Air Force Academy

Chair of Advisory Committee: Dr. Udo W. Pooch

When one speaks of computer science today, it is impossible not to mention networking in the same breath. Networking is changing the way we think, shop, do business, communicate, learn, study, etc. However, one of the main threats to the continued growth of networking is network security. The need for system and network security is well-documented, and dozens of security "tools" already exist to assist system administrators in analyzing the relative vulnerability of their systems. However, these tools are often hard to configure and each operates independent of each other. The SALSA project addresses the need for a system administrator's tool that can provide a variety of applications (ranging from vulnerability scanning to intrusion detection) in a user friendly manner. SALSA incorporates several previously written security applications and it also introduces three new tools. Each of these applications is accessed via the SALSA graphical user interface. Thus, the administrator does need to take time to make the tools work, but can instead spend time analyzing the data the applications provide.

SALSA includes the following security packages: TIGER, COPS, CRACK, SATAN, and SPI-NET. Moreover, SALSA introduces the following applications: SANDS (System And Network Diagnostic Software), PERIOD PLANNER (a crontab tool), and

LOG CUTTERS (an intrusion detection tool). Each of these applications can be launched and configured from the SALSA graphical user interface and there is extensive documentation (included via help buttons) for each of the programs.

SALSA is not a comprehensive computer security tool, for one can not exist. However, it does allow an administrator to more quickly and efficiently secure and maintain a system.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER I

INTRODUCTION

A. General

When one speaks of computer science today, it is impossible not to mention networking in the same breath. Networking is changing the way we think, shop, do business, communicate, learn, study, etc. However, one of the main threats to the continued growth of networking is network security. The need for system and network security is well-documented, and dozens of security "tools" already exist to assist system administrators in analyzing the relative vulnerability of their systems. However, these tools are often hard to configure and each operates independent of each other.

The SALSA project addresses the need for a system administrator's tool that can provide a variety of applications (ranging from vulnerability scanning to intrusion detection) in a user friendly manner. SALSA incorporates several previously written security applications and it also introduces three new tools. Each of these applications is accessed via the SALSA graphical user interface. Thus, the administrator does need to take time to make the tools work, but can instead spend time analyzing the data the applications provide.

SALSA includes the following security packages: TIGER, COPS, CRACK, SATAN, and SPI-NET. Moreover, SALSA introduces the following applications: SANDS (System And Network Diagnostic Software), PERIOD PLANNER (a crontab tool), and LOG CUTTERS (an intrusion detection tool). Each of these applications can be launched and configured from the SALSA graphical user interface and there is extensive documentation (included via help buttons) for each of the programs.

SALSA is not a comprehensive computer security tool, for one can not exist. However,

it does allow an administrator to more quickly and efficiently secure and maintain a system.

B. Overview

This paper will discuss several aspects of the SALSA project. First, the design choices of the application will be presented along with the rationalization and justification of these choices. Following this design discussion each component of the project will be outlined in detail, including its functionality, why it is necessary, and how it is implemented. Third, several potential SALSA extensions and other possible projects will be enumerated. Finally, a brief conclusion will summarize this report.

CHAPTER II

DESIGN DECISIONS

A. Requirements

1. The initial requirements were to create an automated security checker from the AUSCERT security checklist (please see Appendix I). SANDS eventually grew out of this effort.

2. The LOG CUTTERS were added next. The ideas for this tool came from the CERT Coordination Center Generic Security Information checklist also found in Appendix I.

3. Finally, as the project progressed it became apparent that it would be beneficial to have a graphical user interface to each of these tools, and not only to these new tools, but also to previously developed tools (i.e. SATAN, TIGER, etc.).

B. Programming Language

SANDS and the other applications external to the interface were written in C. C was chosen as the programming language for several reasons, which are enumerated below:

1. C's relationship to UNIX. UNIX was written in C, and therefore it made sense to do system checking in C.

2. C has relatively good string manipulation capabilities. This was necessary to accomplish several of the checks.

3. The programmer was already familiar with the programming language. This was a large and relatively difficult project, and programmer expertise was a

prerequisite.

C. The SALSA Interface

It was orginally thought that interface would be written in HTML, and CGI scripts and accessed via a web-browser. However, due to department policy and potential security risks of CGI scripts another option had to be explored.

The interface is written in Tcl/Tk, with several bindings to UNIX commands and the C programs. Tcl/Tk was chosen because of its close ties to C and the UNIX operating system.

CHAPTER III

PROJECT DESCRIPTION

A. SALSA

This is the Security Application Launcher for System Administrators. It was developed to assist System Administrators in securing their systems. It is a collection of tools that are used to evaluate a current system configuration for known security flaws. These are under the heading of "SCANNING AND DIAGNOSTIC TOOLS". Similarly, SALSA has a smaller number of tools that help an administrator determine if an intrusion occurred and when it occurred. These tools fall under the heading "INTRUSION DETECTION TOOLS". Finally, SALSA has a couple of tools that help in the administration of itself and its tools. These include a tool that allows the user to plan and run various tools at the specified time. This is the PERIOD PLANNER. Secondly, there is a tool for determining which users are allowed to access the SALSA interface - this of course is the PASSWORD ADMIN tool.

To run an application, one simply clicks on the button that they desire and there will be additional information available. All of the applications can be launched from within the SALSA interface.

B. SANDS

1. GENERAL INFORMATION

SANDS stands for System and Network Diagnostic Software. It was developed as a diagnostic tool to determine the security state of a given machine based on a security checklist produced by AUSCERT. It is highly configurable, in that it will run any permutation of the security tests based on what the administrator specifies. As noted

earlier, it is a scanning and diagnostic tool that checks compliance of the system verses known vulnerabilities. The following sections provide information on how to configure and run it.

2. CONFIGURING SANDS

SANDS has several switches that can be set by the SALSA interface or by a command line argument. Each of these switches is defined below. It is important to note that a capital letter turns the switch on and a lower case switch turns the switch off. By default, all of the switches are off. It is extremely easy to configure SANDS using these to rules. For example, if I want the bare security check, but in Verbose mode, I would configure it as: ./sands -V. If I wanted a complete check, but avoiding NFS mounted file systems then the command would be: ./sands -Cg. The command ./sands -Ggxv would cause NFS mounts to be pruned, run in non-verbose mode and not check X window security. Each of the switches are defined below:

- -B Turns the files permissions check on. This is described in the section test_file_permissions.c
- -C Turns all switches on (c turns all off).
- -E Shows all of the hosts in hosts.equiv
- -F Does checking for .forward and .exrc files
- -G Does not ignore all NFS mounted file systems (g ignores them)
- -M Tells SANDS to check the majordomo version.
- -N Show the contents of /etc/netgroup
- -P Turns the portmapper check on.

- -R searches for all .rhost files
- -S shows all of the services in inetd.conf
- -V This runs SANDS in verbose mode and this affects the following checks:

  Checks permissions and contents of all .rhosts files

  Checks permissions and contents of all .forward files

  Checks permissions and contents of all .exrc files

  Shows NFS mounts

  Shows more detailed output of services registered with the portmapper.

- -X Checks X window security

3. crondelete.c

This is the subprogram that removes the correct job number from the crontab. It works in conjunction with the cronformatter to number the jobs and then ensures that the correct number is removed. It receives this number from the command line arguments. It is invoked from the PERIOD PLANNER when the REMOVE job option is chosen.

4. cronformatter.c

The cronformatter is used to format the crontab entries in to an intuitive form. It is also used to assign job numbers to the jobs in the crontab. It is invoked whenever the PERIOD PLANNER is invoked and when a change is made to a scheduled job.

5. guard.c

This is used to determine if the username and the SALSA password match the entries in the password file. It is invoked from the login screen.

6. mylast.c

Mylast is used in conjunction with the PAST LOGINS in the LOG CUTTERS. It was the prototype for myreader.c

7. myreader.c

Myreader is used in conjunction with the PAST LOGINS in the LOG CUTTERS. After the "last" command has been executed, my reader "chops through" the log file and "cuts it" to the right output to display the right values to the screen. These values are passed as four command line arguments (the start month, start day, end month, and end day).

8. proj.c

This declares all of the variables and links all of the subprograms in to one big program. It prints out the header, makes sure it has a valid command line argument and sets all of the switches accordingly.

9. runner.c

This is used in conjunction with the SALSA command to view SATAN FACTS. It uses an "awk" command to format the output and display the results. This file was needed because of SALSA's inability to run SATAN with all of its CGI script capability.

10. salsa

This is the Tcl/Tk interface. It creates and manages all of the windows, buttons, menus, screens, tool bars, etc. It requires a login and spawns and runs all other processes.

11. sands.c

This is a simple wrapper that gets the current date and creates a file based on that date and routes the output of proj.c into this file. It also makes sure that it is a valid command line passed in.

12. test_exrc.c

If the -F option is set, then this searches for all .exrc files and lists them. If -V is set it checks the owner and permissions of all of the files that were found.

13. test_file_permissions.c

This checks the permissions and owner of the following files: /etc/passwd /etc/utmp /etc/motd /etc/syslog.pid

It also makes sure that the following directories are owned by root: /etc/ /usr/etc/ /bin /usr/bin /sbin /usr/sbin /tmp (checks that the sticky-bit is set) /var/tmp (checks that the sticky-bit is set)

It also checks that there are only special files in /dev.

If the -B option is set then looks for any special files outside of /dev. It also checks for any world or group writable files or directories on the system. Similarly, it reports all SUID and SGID programs. Finally, it looks for all files owned by bin that are world readable, but not group or world writeable and recommends that these be changed

to ownership of root.

14. test_forward.c

This finds all .forward files. To activate it, the -F option must be set. .forward files should not execute commands they can exploit mailers. If -V is set SANDS reports the permissions and owner of each file.

15. test_hosts.equiv.c

This examines the permissions and owner of /etc/host.equiv file. It also examines the contents of this file to ensure that there is not a "+" by itself, or other unusual characters.

16. test_hosts.equiv.verbose

If the -E option is set then this subprogram shows the contents of hosts.equiv in a formatted output.

17. test_hosts.lpd

This subprogram checks the permissions and owner of host.lpd. It also examines the content of such a file (if it exists) to make sure there are no potential security risks in it.

18. test_inetd.conf.c

This file checks the permissions and owner of /etc/inetd.conf. It also checks to ensure that tftp and rexd are not offered in the services provided by inetd.conf.

19. test_inetd.conf.verbose.c

If the -S option is set then this subprogram shows services offered in inetd.conf.

20. test_majordomo_receive.c

This package reads the mail box and determines the majordomo version.

21. test_majordomo_send.c

This package sends a message to majordomo and requests that the version be sent back.

22. test_netgroup.c

This checks the permissions and owner of /etc/netgroup. If it exists, then SANDS issues a configuration reminder about running NIS or NIS+.

23. test_netgroup.verbose.c

If -N is set, this displays the content of /etc/netgroup.

24. test_nfs.c

If NFS is being used, it checks the permissions of /etc/dfs/dfstab. It also looks at the owner. Next, it scans the contents to make sure that all machine names are fully qualified. Similarly, this subprogram makes sure that the current host does not mount itself.

If the -V option is set, then it displays the current NFS mounts.

25. test_portmapper.c

This program shows all of the services registered with the portmapper. If the -V option is set, then it shows each of the services in a verbose manner (port numbers, etc.).

26. test_rhost.c

If the -R option is set, then SANDS looks for .rhost files. If the -V option is set, it checks the permissions and owner of the file. Also, it checks the contents of each file to make sure that there is not a "+" by itself in each of these files. It also looks for other supicious characters.

27. test_sendmail.c

This makes sure that the current sendmail version is more recent than 8.7.3. If it is a later version, it displays the current version and gives advice to the administrator to make sure that there version is secure.

28. test_services.c

This subprogram simply checks the owner and permissions of /etc/services.

29. test_terminals.c

This checks the permissions and owner of /etc/default/login to make sure that the are set to root, and 644.

30. test_uucp.c

This checks the security of the uucp account. It also looks for any of the uucp subsystem. If there is some of the subsytem left, and if the -B option is selected, then it looks for all files and directories that are owned by uucp and are world writable.

31. test_xsecgen.c

If the -X option is set then it checks the permissions on /tmp to make sure that the sticky-bit is set, and then it looks for all .xinitrc files that contain the entry "xhost +".

32. test_xwin.c

This checks for the version of X11 to make sure that it is atleast version 6. It also checks the version of xdm to make sure that it is more recent than Oct. 1995.

33. unmasker

This tool is not included in the SALSA script, but it is a useful script. It can be used on an /etc/passwd to determine the umask of every user. Each user should have a umask of 77 or 27.

C. THE LOG CUTTERS

This is a basic intrusion detection package. The LOG CUTTERS is comprised of 5 parts. One shows current logins on a machine (using the utmp log). The second one allows the user to view past logins on a machine given a start day and an end day (using the wtmp log). The third part shows current processes on a machine. There is a mechanism for showing active TCP and UDP sockets, and finally, there is a program

for finding modified or accessed files given certain days. These applications each have several features in common:

First, to check a given machine type that machine name into the given entry box. Do NOT put a space after it as this may cause an error.

Also, most of these tools operate by the command "rsh" which is supported in the Texas A&M Department of Computer Science, but can also be a source of security problems. If you run in to errors, then only check the machine where SALSA is located and modify the SALSA script so that the "rsh" commands are removed.

1. CURRENT LOGINS

This package allows the user to determine who is logged into a machine at the given moment. In order to see who is on a machine, the user (identified by the user name used to log into SALSA) must have permission to log on to the target machine. When the tool first comes up, it shows the current logins on the SALSA host machine. This tool uses the utmp file to determine current logins (uses the command who).

2. PAST LOGINS

Past logins are used by administrator to see who logged on to a machine on a given day, on a given machine. As with current logins, the user must have permission to log on to the target machine. When using this tool is important to remember that the logs are implemented as a stack, so the most recent dates are on the top. So, the start date must be the most recent in history. For example, if I wanted to see all of the logins between December 2 and December 10, then I would put the START DATE as December 10 and the END DATE as December 2. This tool uses the wtmp file for the historic logins (uses the command last).

3. NET STATUS

This tool shows the current active ports on a specified machine. It can be configured to show active TCP, UDP sockets, or both. This is done by simply selecting the appropriate radio button. Note, the user must have permission to access the target machine.

An administrator should look for suspicious source destinations. This is just a basic tool and depending on the result will probably require more investigation (uses the command netstat).

4. CURRENT PROCESSES

It may be possible to tamper with log files, but it is much more difficult to mask a process that is running. This tool shows all of the current processes running on a machine. It shows who is "owns" the process, what process it is, and where is was executed from.

When used in conjunction with the other tools it can help give a clearer picture of suspicious activity (uses a formatted ps command).

5. FIND ACCESSED OR CHANGED FILES

Currently, this implementation only allows the user to search for files on the SALSA host. Thus, it is not possible to input a different machine. However, one may specify the top directory from which to begin searching for changed or accessed facts. The radio buttons determine the type of search that is requested (CHANGED or MODIFIED lists files that have had their actual contents changes, where ACCESSED files could just have been read with no changes). Depending on the size of the directory, it may take some time to do the search.

If it is suspected that a break-in has occurred this can be effective for determining where and what the intruder did.

When specifying the days, FROM is the starting date (A date is 24 hours, NOT a calendar day. Thus, all files changed or modified today, a 0 in the input field, means within the last 24 hours not on the calendar day. Two days is 48 hours, etc.) The feild "TO" is then end date. Example, if I want all of the files that have been changed in the /etc/ directory between yesterday and day-before yesterday, then I would specify FROM as 1 and TO as 2 (all files from 1 to 2 days ago). If I want a list of all files changed or modified today then both would be 0. Today to yesterday would be 0,1 in FROM, TO respectively, etc.

D. THE PERIOD PLANNER

The PERIOD PLANNER is used to add jobs to the crontab. This allows and administrator to schedule the following the jobs to run either daily, weekly, or monthly. The jobs that may be scheduled include: When the period planner first comes up, it displays the currently scheduled security jobs. The following jobs can be configured and scheduled and executed:

- SANDS
- TIGER
- COPS
- CRACK

Jobs can be scheduled to run daily, weekly, or monthly. The following sections gives instructions for scheduling and deleting jobs.

1. SCHEDULING A JOB

1. Click on the "ADD JOB" button.
2. Select the "JOB TYPE" from the menu. Notice, that if SANDS or CRACK is selected then there appears a screen to configure these jobs. Please see their respective documentation for instructions on how to configure them.
3. Select the time that the job is to execute from the time menubar.
4. Select the frequency.
5. If the frequency is "WEEKLY" then choose, the day from the subsequent menu that will appear.
6. If the frequency is "MONTHLY" the pick what day of the month that the job will run from the menu that will appear.
7. Once the job is configured correctly, then click the green "COMMIT TO TABLE" button. The PERIOD PLANNER will then schedule the job and redisplay the updated schedule. The "ABORT" button cancels any additions (no job will be added).

2. DELETING A JOB

1. Click on the "REMOVE JOB" button.
2. Input the "Job Number" that is to be removed. NOTE!: The job numbers are not listed in the order they were put into the table. Therefore, the most recently added job, will not have the highest job number. It is important to look at the job number and input it.

3. Click the "DELETE FROM TABLE" button, and the PERIOD PLANNER will then delete the job and redisplay the updated schedule. The "ABORT" button cancels any deletions (no job will be deleted).

E. PASSWORD ADMIN

SALSA uses user names for several functions. This package allows a SALSA user to add new users and passwords, remove users, or change a user's password. It is important to note that these are user-password pairs are valid only for the SALSA system (it is NOT like adding a new user to the /etc/passwd file).

1. ADD A USER

1. In order to add a user to the system, make sure that the "CREATE USER" radio button is selected.

2. Then type in the user's actual unix system user's name. Note this is not an new name, it MUST be the user's system username in order for SALSA to run correctly for that user.

3. Type in the user's SALSA password. It is recommended that this be DIFFERENT than the user's system password. However, it is not required that the two passwords be different. This password may or may not be the same as the system password.

4. Click the green "MAKE CHANGE" button to add the user. Note, if the red "FORGET CHANGE" button is pushed the user will not be added.

5. If the red error "USERNAME ALREADY EXISTS!" appears than the user is already in the system and another user with the same name can not be added.

The user must be changed or deleted.

2. CHANGE A USER'S SALSA PASSWORD

1. To change a user's SALSA password. make sure that the "CHANGE USER" radio button is selected.

2. Type in the user's system name.

3. Type in the user's new SALSA password. There is NO verification of an old password required. Simply type in the new password.

4. Click the green "MAKE CHANGE" button to change the user. Note, if the red "FORGET CHANGE" button is pushed the user will not be changed.

5. If the red error "USERNAME DOES NOT EXIST!" appears than the user is not in the system, and must be added before he/she can be changed.

3. DELETE A USER

1. To delete a user from the SALSA system. Make sure that the "DELETE USER" radio button is selected.

2. Type in the user name. Note, NO password is required to delete a user.

3. Click the green "MAKE CHANGE" button to deleted the user. Note, if the red "FORGET CHANGE" button is pushed the user will not be deleted.

4. If the red error "USERNAME DOES NOT EXIST!" appears than the user is not in the system, and must be added before he/she can be deleted.

F. OTHER TOOLS

1. TIGER

Tiger is also a set of scripts that checks the security of a system against known vulnerabilities. It was developed at Texas A&M and is known for its copious output and it can also take an extended amount of time to run (up to 90 minutes) depending on the type and size of the system.

Tiger and SANDS have similar functions, but their checks are not inclusive. That is to say that they their domains of checking overlap, but are not all inclusive. Please refer to the TIGER documentation for more information.

2. COPS

COPS stands Computer Oracle and Password System. It was developed by Dan Farmer and is similar to TIGER and SANDS. Each of these systems checks known vulnerabilities. However, if an administrator runs each of the tools and cross-references the output of each of the tools it gives a clearer understanding of all of the vulnerabilities and it will be more complete in doing intrusion detection and securing a system. There is extensive COPS documentation is available with the SALSA implementation.

3. CRACK

CRACK is used to check password security. It uses a "dictionary" attack and other standard methods of guessing a user's password. This program should be used to check the security of user's passwords. It is easy to run and does a good job of breaking and reporting the passwords. A run of this can take up to several weeks.

4. SATAN

SATAN stands for "Security Administrator Tool for Analyzing Networks" and is used to report on the network services offered by the target machine. It also gives information about mount points and other security vulnerabilities. SALSA has complete access to the SATAN documentation.

5. SPI-NET

This is a self contained package that was developed by the Department of Energy. It has very good documentation included with it. It helps do a variety of system checks (vulnerability scanning), but also does intrusion detection by monitoring configuration changes.

This is a large, complex system that is very complete, but could almost stand-alone outside of the SALSA suite. Administrators are encouraged to explore all of the capabilities of SPI-NET (i.e. command vs. remote hosts).

CHAPTER IV

EXTENSIONS AND PROJECT IDEAS

The following section enumerates potential extensions to the SALSA system. These extensions range from increasing user friendliness to making the SALSA system more secure and less system dependent. Similarly, there are several other suggestions for potential projects relating to the various applications that are embedded in the SALSA interface.

A. SALSA Extensions/New Features

1. Print Button

The would be a logical addition to the project. It could be implemented in Tcl/Tk and packaged near all of the save buttons. There would need to be some work done between the SALSA interface and the print spool, but this would be a quick and relatively easy addition.

2. Update Job in PERIOD PLANNER

In the current SALSA implemenation there is no way to update a scheduled job. Currently, in order to do an update one must delete the old job and then add the new job with the desired parameters. To implement this addition, a new program would need to be added and cronformatter.c would need to be rewritten. The new program would need to return all of the relevant crontab configuration numbers in a format known a prioi by SALSA. Then, SALSA could easily set the appropriate variables in the interface screen based on the values. The user could update the values and then recommit the job. The old job could be deleted automatically without much

programming effort.

3. Rework of Past Logins in LOG CUTTERS

The Past Logins currently works correctly. However, it is so slow that it is not a practical tool. This could be fixed, but the whole program would need to be reworked.

4. SALSA Password Administration with Encryption

The SALSA usernames and passwords are currently stored in plaintext. This is not insecure because the file is in an obscure location and it is only readable by root. Thus, only someone with root access and who knows where the file is located is able to read it. It would be more secure to use some from of encryption to encode the password file.

5. Replacing rsh Commands

Many of the commands in the the LOG CUTTERS rely on the ability to do the command rsh. However, this command itself is a potential security risk. An option to make SALSA independent of this command would be to write a secure client-server system. Each host in this group could have a server process on a dedicated port. Then, when the SALSA host wanted to get the log information from an individual host, it could connect to the dedicated port, authenticate itself (via PGP, etc.), execute the appropriate command, and get and display the results from the server. The server would then authenticate itself back to the SALSA host. Code for the secure client-server already exists, and it would be a matter of setting up the PGP key rings, configuring the ports and each server, amd then linking it all together in the interface.

6. Help Pages in HTML

The SALSA help pages are all written in plain text. A potentially beneficial addition would be to write all of the help pages in HTML and link them together. This would allow for interactive help and it would be easier to jump between the various help documents. This process would also be a simple matter of writing the basic HTML pages and then cutting and pasting in the text help files. Moreover, a web broweser can be launched directly from the SALSA interface as seen in the SATAN help example.

B. Trend Analysis

Each of these tools produces copious output. The next step in the progression would be to do trend analysis of all of this data. Each of these tools could be used to provide data to an intelligent agent (perhaps an expert system) which could do automated trend analysis and make recommendations, or detect intrusions based on the data that is generated by these applications.

C. Database of Patches

As problems are discovered, the vendor usually issues a "patch" to fix the error. However, over time, there are so many patches it becomes impossible to keep track of the status of all of the different patches. It would be wise to create a current database of all of the required patches to secure a system and then to maintain that database as new patches are issued. It would also need to keep the status of each machine and whether or not it has a given patch installed. It would be particularly useful to have a distributed database that each machine could access to find out if it was in compliance with all of the current patches and then flag the administrator if it was

not update to date and tell them which patch it lacked.

D. Password Cracking Analysis

From running the Crack program, it was noticed that certain passwords were easier to crack than others. Particularly, I noticed that passwords that had a number trailing the word were easier to crack than those that had number in the middle of the word. It would be interesting to do a statistical analysis of the password being cracked to determine some solid metrics for the "quality" of a user password.

CHAPTER V

CONCLUSION

The SALSA project addresses the need for a system administrator's tool that can provide a variety of applications (ranging from vulnerability scanning to intrusion detection) in a user friendly manner. SALSA incorporates several previously written security applications and it also introduces three new tools. Each of these applications is accessed via the SALSA graphical user interface. Thus, the administrator does need to take time to make the tools work, but can instead spend time analyzing the data the applications provide.

This project meets the original design specifications and expanded those requirements to meet other pressing needs. It is a useful system administrators application suite that should be used in by the gold team in the next offering of CPSC 665.

CHAPTER VI

APPENDIX I - SECURITY CHECKLISTS

-----BEGIN PGP SIGNED MESSAGE-----

==========
UNIX Computer Security Checklist (Version 1.1) Last Update 19-Dec-1995
==========

The Australian Computer Emergency Response Team has developed a checklist which assists in removing common and known security vulnerabilities under the UNIX Operating System. It is based around recently discovered security vulnerabilities and other checklists which are readily available (see references in Appendix C).

This document can be retrieved via anonymous ftp from:
ftp://ftp.auscert.org.au/pub/auscert/papers/unix_security_checklist

For information about detecting or recovering from an intrusion, see the CERT security information document which can be retrieved via anonymous ftp from:
ftp://ftp.auscert.org.au/pub/cert/tech_tips/security_info

It is AUSCERT's intention to continue to update this checklist. Any comments should be directed via email to auscert@auscert.org.au. Before using this document, ensure you have the latest version. New versions of this checklist will be placed in the same area on the ftp server and should be checked for periodically.

In order to make effective use of this checklist, readers will need to have a good grasp of basic UNIX system administration concepts. Refer to C.9 and C.10 for books on UNIX system administration.

If possible, apply this checklist to a system before attaching it to a network.

In addition, we recommend that you use the checklist on a regular basis as well as after you install any patches or new versions of the operating system, with consideration given to the appropriateness of each action to your particular situation.

Command examples have been supplied for BSD-like and SVR4-like systems (see Appendix F for operating system details and Appendix G for command details). Full directory paths and program options may vary for different flavours of UNIX. If in doubt, consult your vendor documentation.

For ease of use, the checklist has been organised into separate, logically cohesive sections. All sections are important. An abbreviated version of this checklist can be found in Appendix D.

CHECKLIST INDEX:

APPENDICES:

Any trademarks which appear in this document are registered to their respective owners.

1.0 Patches

* Retrieve the latest patch list from your vendor and install any patches not yet installed that are recommended for your system.
  Some patches may re-enable default configurations. For this reason, it is important to go through this checklist AFTER installing ANY new patches or packages.
* Details on obtaining patches may be found in Section 6.
* Verify the digital signature of any signed files. Tools like PGP may be used to sign files and to verify those signatures.
  (Refer to B.15 for PGP access information).
* If no digital signature is supplied but an md5(1) checksum is supplied, then verify the checksum information to confirm that you have retrieved a valid copy.
  (Refer to B.10 for MD5 access information).
* If only a generic sum(1) checksum is provided, then check that. Be aware that the sum(1) checksum should not be considered secure.

2.0 Network security

The following is a list of features that can be used to help prevent attacks from external sources.

2.1 Filtering

* ENSURE that ONLY those services which are required from outside your domain are allowed through your router filters.
  In particular, if the following are not required outside your domain, then filter them out at the router.

NAME	PORT	PROTOCOL	NAME	PORT	PROTOCOL
echo	7	TCP/UDP	login	513	TCP
systat	11	TCP	shell	514	TCP
netstat	15	TCP	printer	515	TCP
bootp	67	UDP	biff	512	UDP
tftp	69	UDP	who	513	UDP
link	87	TCP	syslog	514	UDP
supdup	95	TCP	uucp	540	TCP
sunrpc	111	TCP/UDP	route	520	UDP
NeWS	144	TCP	openwin	2000	TCP
snmp	161	UDP	NFS	2049	UDP/TCP
xdmcp	177	UDP	X11	6000 to 6000+n	TCP
exec	512	TCP	(where n is the maximum number of X servers you will have)		

Note: Any UDP service that replies to an incoming packet may be subject to a denial of service attack.

See CERT advisory CA-95.01 (C.8) for more details.

Filtering is difficult to implement correctly. For information on packet filtering, please see Firewalls and Internet Security (C.6) and Building Internet Firewalls (C.7).

2.2 "r" commands

2.2.1 If you don't NEED to use the "r" commands...

* DO disable all "r" commands (rlogin, rsh etc.) unless specifically required.

  This may increase your risk of password exposure in network sniffer attacks, but "r" commands have been a regular source of insecurities and attacks. Disabling them is by far the lesser of the two evils (see 2.9.1).

2.2.2 If you must run the "r" commands...

* DO use more secure versions of the "r" commands for cases where there is a specific need.

  Wietse Venema's logdaemon package contains a more secure version of the "r" command daemons. These versions can be configured to consult only /etc/hosts.equiv and not $HOME/.rhosts. There is also an option to disable the use of wildcards ('+').

  Refer to B.13 for access information for the logdaemon package

* DO filter ports 512,513 and 514 (TCP) at the router if you do use any of the "r" commands.

  This will stop people outside your domain from exploiting these commands but will not stop people inside your domain.
  To do this you will need to disable these commands (see 2.2.1).

* DO use tcp_wrappers to provide greater access and logging on these network services (see 2.12).

2.3 /etc/hosts.equiv

2.3.1 It is recommended that the following action be taken whether or not the "r" commands are in use on your system.

* CHECK if the file /etc/hosts.equiv is required.

  If you are running "r" commands, this file allows other hosts to be trusted by your system. Programs such as rlogin can then be used to log on to the same account name on your machine from a trusted machine without supplying a password.
  If you are not running "r" commands or you do not wish to explicitly trust other systems, you should have no use for this file and it should be removed. If it does not exist, it cannot cause you any problems if any of the "r" commands are accidentally re-enabled.

2.3.2 If you must have a /etc/hosts.equiv file

* ENSURE that you keep only a small number of TRUSTED hosts listed.
* DO use netgroups for easier management if you run NIS (also known as YP) or NIS+.
* DO only trust hosts within your domain or under your management.
* ENSURE that you use fully qualified hostnames,
  i.e., hostname.domainname.au
* ENSURE that you do NOT have a '+' entry by itself anywhere in the file as this may allow any user access to the system.
* ENSURE that you do not use '!' or '#' in this file.
  There is no comment character for this file.
* ENSURE that the first character of the file is not '-'.
  Refer to the CERT advisory CA-91:12 (C.8).
* ENSURE that the permissions are set to 600.
* ENSURE that the owner is set to root.
* CHECK it again after each patch or operating system installation.

2.4 /etc/netgroup

* If you are using NIS (YP) or NIS+, DO define each netgroup to contain only usernames or only hostnames.

  All utilities parse /etc/netgroup for either hosts or usernames, but never both. Using separate netgroups makes it

easier to remember the function of each netgroup. The added time required to administer these extra netgroups is a small cost in ensuring that strange permission combinations have not left your machine in an insecure state.
Refer to the manual pages for more information.

2.5 $HOME/.rhosts

2.5.1 It is recommended that the following action be taken whether or not the "r" commands are in use on your system.

* ENSURE that no user has a .rhosts file in their home directory. They pose a greater security risk than /etc/hosts.equiv, as one can be created by each user. There are some genuine needs for these files, so hear each one on a case-by-case basis; e.g., running backups over a network unattended.
* DO use cron to periodically check for, report the contents of and delete $HOME/.rhosts files. Users should be made aware that you regularly perform this type of audit, as directed by policy.

2.5.2 If you must have such a file

* ENSURE the first character of the file is not '-'.
  Refer to the CERT advisory CA-91:12 (C.8).
* ENSURE that the permissions are set to 600.
* ENSURE that the owner of the file is the account's owner.
* ENSURE that the file does NOT contain the symbol "+" on any line as this may allow any user access to this account.
* ENSURE that usage of netgroups within .rhosts does not allow unintended access to this account.
* ENSURE that you do not use '!' or '#' in this file.
  There is no comment character for this file.
* REMEMBER that you can also use logdaemon to restrict the use of $HOME/.rhosts (see 2.2.2).

2.6 NFS

When using NFS, you implicitly trust the security of the NFS server to maintain the integrity of the mounted files.

* DO filter NFS traffic at the router.
  Filter TCP/UDP on port 111
  TCP/UDP on port 2049
  This will prevent machines not on your subnet from accessing file systems exported by your machines.
* DO apply all available patches.
  NFS has had a number of security vulnerabilities.
* DO disable NFS if you do not need it.
  See your vendor supplied documentation for detailed instructions.
* DO enable NFS port monitoring.
  Calls to mount a file system will then be accepted from ports < 1024 only. This will provide added security in some circumstances.
  See your vendor's documentation to determine whether this is an option for your version of UNIX (see also 6.1.8 and 6.2.4).
* DO use /etc/exports or /etc/dfs/dfstab to export ONLY the file systems you need to export.
  If you aren't certain that a file system needs to be exported, then it probably shouldn't be exported.
* DO NOT self-reference an NFS server in its own exports file.
  i.e., The exports file should not export the NFS server to itself in part or in total. In particular, ensure the NFS server is not contained in any netgroups listed in its exports file.
* DO NOT allow the exports file to contain a 'localhost' entry.
* DO export to fully qualified hostnames only.
  i.e., Use the full machine address 'machinename.domainname.au' and

do not abbreviate it to 'machinename'.

* ENSURE that export lists do not exceed 256 characters.
  If you have access lists of hosts within /etc/exports, the list should not exceed 256 characters AFTER any host name aliases have been expanded.
  Refer to the CERT Advisory CA-94:02 (C.8).
* DO run fsirand for all your file systems and rerun it periodically.
  Firstly, ensure that you have installed any patches for fsirand.
  Then ensure the file system is unmounted and run fsirand.
  Predictable file handles assist crackers in abusing NFS.
* ENSURE that you never export file systems unintentionally to the world.
  Use a -access=host.domainname.au option or equivalent in /etc/exports.
  See the manual page for "exports" or "dfstab" for further examples.
* DO export file systems read-only (-ro) whenever possible.
  See the manual page for "exports" or "dfstab" for more information.
* If NIS is required in your situation, then DO use the secure option in the exports file and mount requests (if the secure option is available).
* DO use showmount -e to see what you currently have exported.
* ENSURE that the permissions of /etc/exports are set to 644.
* ENSURE that /etc/exports is owned by root.
* ENSURE that you run a portmapper or rpcbind that does not forward mount requests from clients.
  A malicious NFS client can ask the server's portmapper daemon to forward requests to the mount daemon. The mount daemon processes the request as if it came directly from the portmapper. If the file system is self-mounted this gives the client unauthorised permissions to the file system.
  Refer to section B.14 for how to obtain an alternate portmapper or rpcbind that disallow proxy access.
  Refer to the CERT Advisory 94:15 (C.8).
* REMEMBER that changes in /etc/exports will take effect only after you run /usr/etc/exportfs or equivalent.

Note: A "web of trust" is created between hosts connected to each other via NFS. That is, you are trusting the security of any NFS server you use.

2.7 /etc/hosts.lpd

* ENSURE that the first character of the file is not '-'. (Refer to the CERT advisory CA-91:12 (see C.8)).
* ENSURE that the permissions on this file are set to 600.
* ENSURE that the owner is set to root.
* ENSURE that you do not use '!' or '#' in this file.
  There is no comment character for this file.

2.8 Secure terminals

* This file may be called /etc/ttys, /etc/default/login or /etc/security. See the manual pages for file format and usage information.
* ENSURE that the secure option is removed from all entries that don't need root login capabilities.
  The secure option should be removed from console if you do not want users to be able to reboot in single user mode.
  Note: This does not affect usability of the su(1) command.
* ENSURE that this file is owned by root.
* ENSURE that the permissions on this file are 644.

2.9 Network services

2.9.1 /etc/inetd.conf

* ENSURE that the permissions on this file are set to 600.

* ENSURE that the owner is root.
* DO disable any services which you do not require.
  - To do this we suggest that you comment out ALL services by placing a "#" at the beginning of each line. Then enable the ones you NEED by removing the "#" from the beginning of the line. In particular, it is best to avoid "r" commands and tftp, as they have been major sources of insecurities.
  - For changes to take effect, you need to restart the inetd process. Do this by issuing the commands in G.1. For some systems (including AIX), these commands are not sufficient. Refer to vendor documentation for more information.

2.9.2 Portmapper

* DO disable any non-required services that are started up in the system startup procedures and register with the portmapper. See G.2 for a command to help check for registered services.

2.10 Trivial ftp (tftp)

* If tftp is not needed, comment it out from the file /etc/inetd.conf and restart the inetd process (as above).
* If required, read the AUSCERT Advisory SA-93:05 (see A.1) and follow the recommendations.

2.11 /etc/services

* ENSURE that the permissions on this file are set to 644.
* ENSURE that the owner is root.

2.12 tcp_wrapper (also known as log_tcp)

* ENSURE that you are using this package.
  - Customise and install it for your system.
  - Enable PARANOID mode
  - Consider running with the RFC931 option
  - Deny all hosts by putting "all:all" in /etc/hosts.deny and explicitly list trusted hosts who are allowed access to your machine in /etc/hosts.allow.
  - See the documentation supplied with this package for details about how to do the above.
* DO wrap all TCP services that you have enabled in /etc/inetd.conf
* DO consider wrapping any udp services you have enabled. If you wrap them, then you will have to use the nowait option in the /etc/inetd.conf file.
* See section B.4 for instructions to obtain tcp_wrapper.

2.13 /etc/aliases

* Comment out the "decode" alias by placing a "#" at the beginning of the line. For this change to take effect you will need to run /usr/bin/newaliases. If you run NIS (YP), you will then need to rebuild your maps (see G.3).
* ENSURE that all programs executable by an alias are owned by root, have permissions 755 and are stored in a systems directory e.g., /usr/local/bin. If smrsh is in use, program execution may be further restricted. Refer to the smrsh documentation for more details (see B.9).

2.14 Sendmail

* DO use the latest version of Eric Allman's sendmail 8.x (currently 8.7.3), as it currently contains no KNOWN vulnerabilities.
  The latest version is available via anonymous FTP from:
  ftp://ftp.auscert.org.au/pub/mirrors/ftp.cs.berkeley.edu/ucb/sendmail
  NOTE: If you don't already run Eric Allman's sendmail.8.7.*,

then it may take you some time to build, install, and configure the system to your needs. Other sendmail(8) configuration files may not be compatible with sendmail(8) 8.7.x. There is some help available for converting from SUN's sendmail: bundled with the distribution of sendmail(8) v8.7.x is a document on converting standard SUN configuration files to sendmail(8) v8.*. This is located in the distribution, in the file:
contrib/converting.sun.configs

* If you use a vendor version of sendmail, ENSURE that you have installed the latest patches as sendmail(8) has been a source of a number of security vulnerabilities.
  Refer to AUSCERT Advisories SA-93:10, AA-95.08 and AA-95.09b (A.1) and CERT Advisories CA-94:12, CA-95:05 and CA-95:08 (C.8).
* If you require progmailer functionality then DO use smrsh (see B.9).
* If you do not require progmailer functionality then DO disable mail to programs by setting this field to /bin/false in the sendmail configuration file.
* ENSURE that your version of sendmail does not have the wizard password enabled (see G.4). ENSURE that if you have a line starting with "OW" in /etc/sendmail.cf, it only has a "*" next to it.
* DO increase sendmail(8) logging to a minimum log level of 9.
  This will help detect attempted exploitation of the sendmail(8) vulnerabilities. See G.5 for example commands.
* DO increase the level of logging provided by syslog.
  Enable a minimum level of "info" for mail messages to be logged to the console and/or the syslog file. See G.6 for example code and instructions.
* REMEMBER that you will need to restart sendmail for any changes to take effect. If you are running a frozen configuration file (sendmail.fc), you will need to rebuild it before restarting sendmail(8) (see G.7).

2.15 majordomo

* ENSURE that your version is greater than 1.91.
  See AUSCERT Advisory SA-94.03 (see A.1) for more details.

2.16 fingerd

* If your version of fingerd is older than than 5 November 1988, DO replace it with a newer version.
* Finger can provide a would-be intruder with a lot of information about your host. CONSIDER the finger information you provide and think about reducing the content by disabling finger or by replacing it with a version that only offers restricted information.
  NOTE: other services such as rusers and netstat may give out similar information.
* DO NOT use GNU finger v1.37 as it may allow intruders to read any file.

2.17 UUCP

* DO disable the uucp account, including the shell that it executes for logging in, if it is not used at your site.
  uucp may be shipped in a dangerous state.
* REMOVE any .rhosts file at the uucp home directory.
* ENSURE that the file L.cmds is owned by root.
* ENSURE that no uucp owned files or directories are world writable.
* ENSURE that you have assigned a different uucp login for each site that needs uucp access to your machine.
* ENSURE that you have limited the number of commands that each uucp login can execute to a bare minimum.
* DO consider deleting the whole uucp subsystem if it is not required.
* ENSURE there are no vendor-supplied uucp or root crontab entries.

2.18 REXD

* DO disable this service.
  Comment this out in the inetd.conf file. See section 2.9.1 for details on how to do this. rexd servers have little or no security in their design or implementation. Intruders can exploit this service to execute commands as any user.

2.19 World Wide Web (WWW) - httpd

* ENSURE that you are using the most recent version of the http daemon of your choice.
* DO run the server daemon httpd as a specially created nonprivileged user such as 'httpd'.
  This way, if an intruder finds a vulnerability in the server they will only have access privileges for this unprivileged user.
* DO NOT run the server daemon as root.
* DO NOT run the client processes as root.
* DO run httpd in a chroot(1) environment.
  This sets up an alternate root directory that severely limits access of http clients to the rest of the disk.
* For systems which do not have a chroot(1) command, use of chrootuid (see B.16) may be of assistance.
* DO carefully go through the configuration options for your server.
* DO use the configuration options to give extra protection to sensitive directories by turning off the 'include files' feature.
  This will disallow files from these directories from being included in HTML documents.
* DO use CGIWRAP. (See B.17)
* DO NOT run CGI (Common Gateway Interface) scripts if they are not required.
* DO be very careful in constructing CGI programs.
  These programs compute information to be returned to clients and are often driven by input from the remote user who may be hostile. If these programs are not carefully constructed, it may be possible for remote users to subvert them to execute arbitrary commands on the server system. Almost all vulnerabilities arise from these issues.
* DO provide CGIs as statically linked binaries rather than as interpreted scripts.
  This will remove the need for a command interpreter to be available inside the chrooted environment.
* ENSURE that the contents, permissions and ownership of files in the cgi-bin directory are what you expect them to be (see your site security policy document for more details).
* AVOID passing user input directly to command interpreters such as Perl, AWK, UNIX shells or programs that allow commands to be embedded in outgoing messages such as /usr/ucb/mail
* FILTER user input for potentially dangerous characters before it is passed to any command interpreters.
  Possibly dangerous characters include \n \r (.,/;~!)>|`&$'< .
  (Refer to the CERT Advisory CA-95:04 (see C.8)).

3.0 ftpd and anonymous ftp

3.1 Versions

* ENSURE that you are using the most recent version of the ftp daemon of your choice.
* DO consider installing the Washington University ftpd if you don't already have it (see B.19).
* For BSDI systems, patch 005 should be applied to version 1.1 of the BSD/386 software (see B.20).

3.2 Configuration

* CHECK all default configuration options on your ftp server.
* ENSURE that your ftp server does not have the SITE EXEC command (see G.8 for command details).
* ENSURE that you have set up a file /etc/ftpusers which specifies those users that are NOT allowed to connect to your ftpd.
  This should include, as a MINIMUM, the entries: root, bin, uucp, ingres, daemon, news, nobody and ALL vendor supplied accounts.

3.3 Anonymous ftp only

* To ascertain whether you are running anonymous ftp, try to connect to the localhost using anonymous ftp. Be sure to give an RFC822 compliant username as the password (see G.9).
* To disable anonymous ftp, move or delete all files in ~ftp/ and then remove the user ftp from your password file.
* If you are running distributed passwords (e.g., NIS, NIS+) then you will need to check the password entries served to your machine as well as those in your local password file.

3.3.1 Configuration of your ftp server

* CHECK all default configuration options on your ftp server.
  Not all versions of ftp are configurable. If you have a configurable version of ftp (e.g., wu-ftp) then make sure that all delete, overwrite, rename, chmod and umask options (there may be others) are NOT allowed for guests and anonymous users. In general, anonymous users should not have any unnecessary privileges.
* ENSURE that you DO NOT include a command interpreter (such as a shell or tools like perl) in ~ftp/bin, ~ftp/usr/bin, ~ftp/sbin or similar directory configurations that can be executed by SITE EXEC (Refer to AUSCERT advisory SA-94.01 (see A.1)).
* DO NOT keep system commands in ~ftp/bin, ~ftp/usr/bin, ~ftp/sbin or similar directory configurations that can be executed by SITE EXEC.
  It may be necessary to keep some commands, such as uncompress, in these locations. Consider the inclusion of each command on a case by case basis and be aware that the presence of such commands may make it possible for local users to gain unauthorised access. Be wary of including commands that can execute arbitrary commands. For example, some versions of tar may allow you to execute an arbitrary file.
  (Refer to AUSCERT advisory SA-94.01 (see A.1)).
* ENSURE that you use an invalid password and user shell for the ftp entry in the system password file and the shadow password file (if you have one). It should look something like:
  ftp:*:400:400:Anonymous FTP:/home/ftp:/bin/false
  where /home/ftp is the anonymous ftp area.
* ENSURE that the permissions of the ftp home directory (~ftp/) are set to 555 (read nowrite execute), owner set to root (NOT ftp).
* ENSURE that you DO NOT have a copy of your real /etc/passwd file as ~ftp/etc/passwd.
  Create one from scratch with permissions 444, owned by root. It should not contain the names of any accounts in your real password file. It should contain only root and ftp. These should be dummy entries with disabled passwords eg:
  root:*:0:0:Ftp maintainer::
  ftp:*:400:400:Anonymous ftp::
  The password file is used only to provide uid to username mapping for ls(1) listings.
* ENSURE that you DO NOT have a copy of your real /etc/group file as ~ftp/etc/group.

Create one from scratch with permissions 444, owned by root.

* ENSURE the files ~ftp/.rhosts and ~ftp/.forward do not exist.
* DO set the login shell of the ftp account to a non-functional shell such as /bin/false.

3.3.2 Permissions

* ENSURE NO files or directories are owned by the ftp account or have the same group as the ftp account.
  If they are, it may be possible for an intruder to replace them with a trojan version.
* ENSURE that the anonymous ftp user cannot create files or directories in ANY directory unless required (see Section 3.3.3).
* ENSURE that the anonymous ftp user can only read information in public areas.
* ENSURE that the permissions of the ftp home directory (~ftp/) are set to 555 (read nowrite execute), owner set to root (NOT ftp).
* ENSURE that the system subdirectories ~ftp/etc and ~ftp/bin have the permissions 111 only, owner set to root.
* ENSURE that the permissions of files in ~ftp/bin/* have the permissions 111 only, owner set to root.
* ENSURE that the permissions of files in ~ftp/etc/* are set to 444, owner set to root.
* ENSURE that there is a mail alias for ftp to avoid mail bounces.
* ENSURE /usr/spool/mail/ftp is owned by root with permissions 400.

3.3.3 Writable directories

* ENSURE that you don't have any writable directories.
  It is safest not to have any writable directories. If you do have any, we recommend that you limit the number to one.
* ENSURE that writable directories are not also readable.
  Directories that are both writable and readable may be used in an unauthorised manner.
* ENSURE that any writable directories are owned by root and have permissions 1733.
* DO put writable directories on a separate partition if possible.
  This will help to prevent denial of service attacks.
* DO read Anonymous FTP Configuration Guidelines (see B.21).

3.3.4 Disk mounting

* NEVER mount disks from other machines to the ~ftp hierarchy unless they are set read-only in the mount command.

4.0 Password and account security

This section of the checklist can be incorporated as part of a password and account usage policy.

4.1 Policy

* ENSURE that you have a password policy for your site.
  See the AUSCERT Advisory SA-93.04 (see A.1).
* ENSURE you have a User Registration Form for each user on each system. Make sure that this form includes a section that the intending applicant signs, stating that they have read your account usage policy and what the consequences are if they misuse their account.

4.2 Proactive Checking

* DO use anlpasswd to proactively screen passwords as they are entered.
  This program runs a series of checks on passwords when they are set, which assists in avoiding poor passwords. It works with

normal, shadow and NIS (or yp) password systems.
(Refer to section B.3 for how to obtain it).

* DO check passwords periodically with Crack.
(Refer to section B.1 for how to obtain Crack).
* DO apply password ageing (if possible).

4.3 NIS, NIS+ and /etc/passwd entries

* DO NOT run NIS or NIS+ if you don't really need it.
* If NIS functionality is required, DO use NIS+ if possible.
* ENSURE that the only machines that have a '+' entry in the /etc/passwd files are NIS (YP) clients; i.e., NOT the NIS master server!
There appears to be conflicting documentation and implementations regarding the '+' entry format and so a generic solution is not available here. It would be best to consult your vendor's documentation.
Some of the available documentation suggests placing a '*' in the password field, which is NOT consistent across all implementations of NIS. We recommend testing your systems on a case-by-case basis to see if they correctly implement the '*' in the password field.
See G.10 for instructions.
* ENSURE that /etc/rc.local or the equivalent startup procedure is set up to start ypbind with the -s option.
This may not be applicable on all systems. Check your documentation.
* DO use secure RPC.

4.4 Password shadowing

* DO enable vendor supplied password shadowing or a third party product.
Password shadowing restricts access to users' encrypted passwords.
* DO periodically audit your password and shadow password files for unauthorised additions or inconsistencies.

4.5 Administration

* ENSURE that you regularly audit your system for dormant accounts and disable any that have not been used for a specified period, say 3 months. Send out account renewal notices by post and delete any accounts of users that do not reply.
[NOTE: Do not email renewal notices because any accounts being used illegitimately will reply as expected and hence will not be discovered]
* ENSURE that all accounts have passwords. Check shadow or NIS passwords too, if you have them.
i.e., the password field is not empty.
* ENSURE that any user area is adequately backed up and archived.
* DO regularly monitor logs for successful and unsuccessful su(1) attempts.
* DO regularly check for repeated login failures.
* DO regularly check for LOGIN REFUSED messages.
* Consider quotas on user accounts if you do not have them.
* Consider requiring that users physically identify themselves before granting any requests regarding accounts (e.g., before creating a user account).

4.6 Special accounts

* ENSURE that there are no shared accounts other than root in accordance with site security policy.
i.e., more than one person should not know the password to an account.
* Disable guest accounts.
Better yet, do not create guest accounts!

[NOTE: Some systems come preconfigured with guest accounts]

* DO use special groups (such as the "wheel" group under SunOS) to restrict which users can use su to become root.
* DISABLE ALL default vendor accounts shipped with the Operating System. This should be checked after each upgrade or installation.
* DO Disable accounts that have no password which execute a command, for example "sync".
  Delete or change ownership of any files owned by these accounts. Ensure that these accounts do not have any cron or at jobs. It is best to remove these accounts entirely.
* DO assign non-functional shells (such as /bin/false) to system accounts such as bin and daemon and to the sync account if it is not needed.
* DO put system accounts in the /etc/ftpusers file so they cannot use ftp.
  This should include, as a MINIMUM, the entries: root, bin, uucp, ingres, daemon, news, nobody and ALL vendor supplied accounts.

4.7 Root account

* DO restrict the number of people who know the root password. These should be the same users registered with groupid 0 (e.g., wheel group on SunOS). Typically this is limited to at most 3 or 4 people.
* DO NOT log in as root over the network, in accordance with site security policy.
* DO su from user accounts rather than logging in as root. This provides greater accountability.
* ENSURE root does not have a ~/.rhosts file.
* ENSURE "." is not in root's search path.
* ENSURE root's login files do not source any other files not owned by root or which are group or world writable.
* ENSURE root cron job files do not source any other files not owned by root or which are group or world writable.
* DO use absolute path names when root.
  e.g., /bin/su, /bin/find, /bin/passwd. This is to stop the possibility of root accidentally executing a trojan horse. To execute commands in the current directory, root should prefix the command with "./", e.g., ./command.

4.8 .netrc files

* DO NOT use .netrc files unless it is absolutely necessary.
* If .netrc files must be used, DO NOT store password information in them.

4.9 GCOS field

* DO include information in the GCOS field of the password file which can be used to identify your site if the password file is stolen.
  e.g., joe:*:10:10:Joe Bloggs, Organisation X:/home/joe:/bin/sh

5.0 File system security

5.1 General

* ENSURE that there are no .exrc files on your system that have no legitimate purpose.
* DO consider using the EXINIT environment variable to disable .exrc file functionality.
  These files may inadvertently perform commands that may compromise the security of your system if you happen to start either vi(1) or ex(1) in a directory which contains such a file.

See G.11 for example commands to find .exrc files.

* ENSURE that any .forward files in user home directories do not execute an unauthorised command or program.
  The mailer may be fooled into allowing a normal user privileged access. Authorised programs may be restricted through use of smrsh (see B.9).
  See G.12 for example commands to find .forward files.
  (Refer to AUSCERT Advisory SA-93.10 (see A.1)).

5.2 Startup and shutdown scripts

* ENSURE startup and shutdown scripts do not chmod 666 motd.
  This allows users to change system message for the day.
* ENSURE that the line "rm -f /tmp/t1" (or similar) exists in a startup script to clean up the temporary file used to create /etc/motd. This should occur BEFORE the code to startup the local daemons.

5.3 /usr/lib/expreserve

* DO replace versions of /usr/lib/expreserve prior to July 1993 with a recommended patch from your vendor.
  If this is not possible, then remove execute permission on /usr/lib/expreserve (see G.13).
  This will mean that users who edit their files with either vi(1) or ex(1) and have their sessions interrupted, will not be able to recover their lost work. If you implement the above workaround, please advise your users to regularly save their editing sessions.
  (Refer to the CERT advisory CA-93:09 for advice on fixing this problem for the SunOS and Solaris environments).

5.4 External file systems/devices

* DO mount file systems non-setuid and read-only where practical.
  (Refer to section 2.6)

5.5 File Permissions

* ENSURE that the permissions of /etc/utmp are set to 644.
* ENSURE that the permissions of /etc/sm and /etc/sm.bak are set to 2755.
* ENSURE that the permissions of /etc/state are set to 644.
* ENSURE that the permissions of /etc/motd and /etc/mtab are set to 644.
* ENSURE that the permissions of /etc/syslog.pid are set to 644.
  [NOTE: this may be reset each time you restart syslog.]
* DO consider removing read access to files that users do not need to access.
* ENSURE that the kernel (e.g., /vmunix) is owned by root, has group set to 0 (wheel on SunOS) and permissions set to 644.
* ENSURE that /etc, /usr/etc, /bin, /usr/bin, /sbin, /usr/sbin, /tmp and /var/tmp are owned by root and that the sticky-bit is set on /tmp and on /var/tmp (see G.14). Refer to the AUSCERT Advisory AA-95:05 (see A.1).
* ENSURE that there are no unexpected world writable files or directories on your system.
  See G.15 for example commands to find group and world writable files and directories.
* CHECK that files which have the SUID or SGID bit enabled, should have it enabled (see G.16).
* ENSURE the umask value for each user is set to something sensible like 027 or 077.
  (Refer to section E.1 for a shell script to check this).
* ENSURE all files in /dev are special files.
  Special files are identified with a letter in the first position of the permissions bits. See G.17 for a command to find files in /dev which are not special files or directories.
  Note: Some systems have directories and a shell script in /dev which

may be legitimate. Please check the manual pages for more information.

* ENSURE that there are no unexpected special files outside /dev. See G.18 for a command to find any block special or character special files.

5.6 Files run by root

AUSCERT recommends that anything run by root should be owned by root, should not be world or group writable and should be located in a directory where every directory in the path is owned by root and is not group or world writable.

* CHECK the contents of the following files for the root account. Any programs or scripts referenced in these files should meet the above requirements:
  - ~/.login, ~/.profile and similar login initialisation files
  - ~/.exrc and similar program initialisation files
  - ~/.logout and similar session cleanup files
  - crontab and at entries
  - files on NFS partitions
  - /etc/rc* and similar system startup and shutdown files
* If any programs or scripts referenced in these files source further programs or scripts they also need to be verified.

5.7 Bin ownership

Many systems ship files and directories owned by bin (or sys). This varies from system to system and may have serious security implications.

* CHANGE all non-setuid files and all non-setgid files and directories that are world readable but not world or group writable and that are owned by bin to ownership of root, with group id 0 (wheel group under SunOS 4.1.x).
  - Please note that under Solaris 2.x changing ownership of system files can cause warning messages during installation of patches and system packages.
  - Anything else should be verified with the vendor.

5.8 Tiger/COPS

* Do run one or both of these. Many of the checks in this section can be automated by using these programs.
* To obtain these programs, see B.2.

5.9 Tripwire

* DO run statically linked binary
* DO store the binary, the database and the configuration file on hardware write-protected media.
* To obtain this program, see B.5.

6.0 Vendor operating system specific security

The following is a list of security issues that relate to specific UNIX operating systems. This is not necessarily a complete list of available UNIX types or of problems for those that are listed.

6.1 SunOS 4.1.x

6.1.1 Patches

* DO regularly ask your vendor for a complete list of patches. Sun regularly updates a list of recommended and security patches, which is available from:
  ftp://ftp.auscert.org.au/pub/mirrors/sunsolve1.sun.com/*

or
ftp://sunsolve1.sun.com/pub/patches/*

6.1.2 IP forwarding and source routing

This is particularly relevant if you are using your SUN box as a bastion host or duel homed system.

* ENSURE IP forwarding is disabled.
  You will need the following line in the kernel configuration file:
  options "IPFORWARDING=-1"
  For information on how to customise a kernel, see the file:
  /usr/sys/'arch'/conf/README
* DO also consider disabling source routing.
  Leaving source routing enabled may allow unauthorised traffic through. Unfortunately there is no official method or patch for turning source routing off. There is however an unsupported patch. It is available via anonymous ftp from
  ftp://ftp.auscert.org.au/pub/mirrors/ftp.greatcircle.com/v03.n153.Z

6.1.3 Framebuffers /dev/fb

If somebody can log in to your Sun workstation from a remote source, they can read the contents of your Framebuffer, which is /dev/fb. Sun provides a mechanism which allows the user logging in on the console to have exclusive access to the Framebuffer, by using the file /etc/fbtab.
A sample /etc/fbtab file:

```
#
# File:          /etc/fbtab
# Purpose:       Specifies that upon login to /dev/console, the
#                owner, group and permissions of all supported
#                devices, including the framebuffer, will be set to
#                the user's username, the user's group and 0600.
# Comments:      SunOS specific.
# Note:          You cannot use \ to continue a line.
#
# Format:
# Device         Permission      Colon separated device list.
#
/dev/console     0600            /dev/fb
/dev/console     0600            /dev/bwone0:/dev/bwtwo0
/dev/console     0600            /dev/cgone0:/dev/cgtwo0:/dev/cgthree0
/dev/console     0600            /dev/cgfour0:/dev/cgsix0:/dev/cgeight0
/dev/console     0600            /dev/cgnine0:/dev/cgtwelve0
#
/dev/console     0600            /dev/kb:/dev/mouse
/dev/console     0600            /dev/fd0c:/dev/rfd0c
```

After the above file has been created, reboot your machine, or log out fully, then log back in again.
Read the man page for fbtab(5) for more information.

* The login replacement from Wietse Venema's logdaemon package supports a similar feature.
  (Refer to B.13 for information on how to retrieve the logdaemon package)

6.1.4 /usr/kvm/sys/*

* ENSURE all files and directories under /usr/kvm/sys/ are not writable by group.
  In SunOS 4.1.4 the default mode is 2775 with group staff, allowing users in group staff to trojan the kernel.

6.1.5 /usr/kvm/crash

* REMOVE setgid privileges on /usr/kvm/crash with the command:

```
# /bin/chmod g-s /usr/kvm/crash
```

A group of kmem allows users to read the virtual memory of a running system.

6.1.6 /dev/nit (Network Interface Tap)

* DO run the CERT tool cpm to check if your system is running in promiscuous mode.
  For access details for cpm see B.6.
* DO disable the /dev/nit interface if you do not need to run in promiscuous mode.
  - For SunOS 4.x and Solbourne systems, the promiscuous interface to the network can be eliminated by removing the /dev/nit capability from the kernel. Once the procedure is complete, you may remove the device file /dev/nit since it is no longer functional.
  - Apply "method 1" as outlined in the System and Network Administration manual, in the section, "Sun System Administration Procedures," Chapter 9, "Reconfiguring the System Kernel." Excerpts from the method are reproduced below:

```
# cd /usr/kvm/sys/sun[3,3x,4,4c]/conf
# cp CONFIG_FILE SYS_NAME
```

[NOTE: that at this step, you should replace the CONFIG_FILE with your system specific configuration file if one exists.]

```
# chmod +w SYS_NAME
# vi SYS_NAME
#
# The following are for streams NIT support.  NIT is used by
# etherfind, traffic, rarpd, and ndbootd.  As a rule of thumb,
# NIT is almost always needed on a server and almost never
# needed on a diskless client.
#
pseudo-device   snit            # streams NIT
pseudo-device   pf              # packet filter
pseudo-device   nbuf            # NIT buffering module
```

[Comment out the 3 "pseudo-device" lines; save and exit the editor before proceeding.]

```
# config SYS_NAME
# cd ../SYS_NAME
# make
# mv /vmunix /vmunix.old
# cp vmunix /vmunix
# /etc/halt
> b
```

[This step will reboot the system with the new kernel.]
[NOTE: that even after the new kernel is installed, you need to take care to ensure that the previous vmunix.old , or other kernel, is not used to reboot the system.]

See CERT Advisory CA_94.01 (see C.8)

6.1.7 Loadable drivers option

* DO remove the option for loadable modules from the kernel.
  This will mean that a rebuild of the kernel and a reboot will be necessary in order to load any additional kernel modules and intruders will be prevented from being able to load extra kernel modules dynamically. To remove this option, comment out the line

```
options         VDDRV           # loadable modules
```

from the kernel configuration file and re-compile the kernel.

NOTE: Some software may expect to be able to load additional modules such as device drivers.

NOTE: Even after the new kernel is installed, you need to take care to ensure that the previous vmunix.old , or other kernel, is not used to reboot the system.

6.1.8 NFS port monitoring

* DO enable NFS port monitoring (see also section 2.6).

Add the following commands to /etc/rc.local:

```
/bin/echo "nfs_portmon/W1" | /bin/adb -w /vmunix /dev/kmem > \
  /dev/null 2>&1
rpc.mountd
```

6.2 Solaris 2.x

6.2.1 Patches

* DO regularly ask your vendor for a complete list of patches. Sun regularly updates a list of recommended and security patches, which is available from:

ftp://ftp.auscert.org.au/pub/mirrors/sunsolve1.sun.com/*

or

ftp://sunsolve1.sun.com/pub/patches/*

6.2.2 IP forwarding and source routing

This is particularly relevant if you are using your SUN box as a bastion host or duel homed system.

* DO disable IP forwarding and source routing.

To do this you will need to edit the file /etc/rc.2.d/S69.inet and set the options ip_forwarding and ip_ip_forward_src_routed to zero as illustrated below:

```
ndd -set /dev/ip ip_forwarding 0
ndd -set /dev/ip ip_ip_forward_src_routed 0
```

For the changes to take effect you will then need to reboot.

6.2.3 Framebuffers /dev/fbs

* Solaris versions 2.3 and above have a protection facility for framebuffers which is a superset of the functionality provided by /etc/fbtab in SunOS 4.1.x.

* Under Solaris, /dev/fbs is a directory that contains links to the framebuffer devices. The /etc/logindevperm file contains information that is used by login(1) and ttymon(1M) to change the owner, group, and permissions of devices upon logging into or out of a console device. By default, this file contains lines for the keyboard, mouse, audio, and frame buffer devices.

A sample /etc/logindevperm file:

```
#
# File:     /etc/logindevperm
# Purpose:  Specifies that upon login to /dev/console, the
#           owner, group and permissions of all supported
#           devices, including the framebuffer, will be set to
#          the user's username, the user's group and 0600.
# Comments:        SunOS specific.
# Note:    You cannot use \ to continue a line.
#
# Format:
# Device    Permission     Colon separated device list.
#
/dev/console   0600        /dev/kbd:/dev/mouse
/dev/console   0600        /dev/sound/*          # audio devices
/dev/console   0600        /dev/fbs/*            # frame buffers
```

Read the man page for logindevperm(4) for more information.

6.2.4 NFS port monitoring

* DO enable NFS port monitoring.
  To do this add the following lines to /etc/system:
  ```
  set nfs:nfs_portmon = 1
  ```
  or in Solaris version 2.5
  ```
  set nfssrv:nfs_portmon = 1
  ```
* See also section 2.6.

6.3 IRIX

* DO regularly ask your vendor for a complete list of patches.
* Some IRIX patches are available via anonymous ftp from:
  ftp://ftp.auscert.org.au/pub/mirrors/ftp.sgi.com/security/*
  or
  ftp://ftp.auscert.org.au/pub/mirrors/sgigate.sgi.com/*
* DO read the FAQ on IRIX security.
  A copy can be obtained via anonymous ftp from
  ftp://ftp.auscert.org.au/pub/mirrors/ftp.uu.net/sgi/security.Z
* For systems which do not have the chroot(1) command, use of chrootuid (see B.16) may be of assistance.
* DO use the software tool rscan.
  It checks for many common IRIX-specific security vulnerabilities and problems. (Refer to B.11 for information on where to get a copy of rscan)

6.4 AIX

* DO regularly ask your vendor for a complete list of patches.
* Some AIX patches are available via anonymous ftp from:
  ftp://ftp.auscert.org.au/pub/mirrors/software.watson.ibm.com
  /aix-patches

6.5 HP/UX

* DO regularly ask your vendor for a complete list of patches.
  HP has set up an automatic server to allow patches and other security information to be retrieved via email. Email should be sent to the address
  ```
  support@support.mayfield.hp.com.
  ```
  The subject line of the message will be ignored. The body (text) of the message should be of the format
  ```
  send XXXX
  ```
  where XXXX is the identifier for the information you want retrieved. For example, to retrieve the patch PHSS_4834, the message would be
  ```
  send PHSS_4834.
  ```
  To receive the HP SupportLine mail service user's guide
  ```
  send guide.txt
  ```
  To receive the readme file for a patch
  ```
  send doc PHSS_4834
  ```
  To receive the original HP bulletin
  ```
  send doc HPSBUX9410-018.
  ```
  HP also has a World Wide Web server to browse and retrieve bulletins and patches. The URL is:
  ```
  http://support.mayfield.hp.com/
  ```

6.6 OSF

* DO regularly ask your vendor for a complete list of patches.
  Some patches are available from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.service.digital.com
/osf//ssrt*
or
ftp://ftp.service.digital.com/pub/osf//ssrt*
where is the version of the operating system that you run.

6.7 ULTRIX

* DO regularly ask your vendor for a complete list of patches.
Some patches are available from:
ftp://ftp.auscert.org.au/pub/mirrors/ftp.service.digital.com
/ultrix/
//ssrt*
or
ftp://ftp.service.digital.com/pub/ultrix/
//ssrt*
where
is either mips or vax; and
where is the version of the operating system that you run.

7.0 Security and the X Window System

Access to your X server may be controlled through either a host-based or user-based method. The former is left to the discretion of the Systems Administrator at your site and is useful as long as all hosts registered in the /etc/Xn.hosts file have users that can be trusted, where "n" represents your X server's number.

This may not be possible at every site, so a better method is to educate each and every user about the security implications (see references below). Better still, when setting up a user, give them a set of X security related template files, such as .xserverrc and .xinitrc. These are located in the users home directory.

You are strongly advised to read the section on X window system security referred to in the X Window System Administrators Guide (C.4).

7.1 Problems with xdm

Note: Release 6 of X11 is now available and solves many problems associated with X security which were present in previous releases. If possible, obtain the source for R6 and compile and install it on your system. See B.18 for how to retrieve the source for X11R6.

* xdm bypasses the normal getty and login functions, which means that quotas for the user, ownership of /dev/console and possibly other preventive measures put in place by you may be ignored.
* You should consult your vendor and ask about potential security holes in xdm and what fixes are available.
* If you are running a version of xdm earlier than October 1995 then you should update to a newer version.
(Refer to CERT Vendor-Initiated Bulletin VB-95:08, see C.8)

7.2 X security - General

* DO Read the man pages for xauth and Xsecurity.
Use this information to set up the security level you require.
* ENSURE that the permissions on /tmp are set to 1777 (or drwxrwxrwt).
i.e., the sticky bit should be set. The owner MUST always be root and group ownership should be set to group-id 0, which is "wheel" or "system".
  - If the sticky bit is set, no one other than the owner can delete the file /tmp/.X11-unix/X0, which is a socket for your X server. Once this file is deleted, your X server will no

longer be accessible.
- See G.14 for example commands to set the correct permissions and ownership for /tmp.

* DO use the X magic cookie mechanism MIT-MAGIC-COOKIE-1 or better.
  With logins under the control of xdm (see 7.1), you can turn on authentication by editing the xdm-config file and setting the DisplayManager*authorize attribute to true.
  When granting access to the screen from another machine, use the xauth command in preference to the xhost command.
* DO not permit access from arbitrary hosts.
  Remove all instances of the 'xhost +' command from the system-wide Xsession file, from user .xsession files, and from any application programs or shell scripts that use the X window system.

==========

Appendix A: Other AUSCERT information sources

A.1 AUSCERT advisories and alerts

Past AUSCERT advisories and alerts can be retrieved via anonymous ftp from

ftp://ftp.auscert.org.au/pub/auscert/advisory/

A.2 AUSCERT's World Wide Web server

AUSCERT maintains a World Wide Web server. Its URL is

http://www.auscert.org.au

A.3 AUSCERT's ftp server

AUSCERT maintains an ftp server with an extensive range of tools and documents. Please browse through it. Its URL is

ftp://ftp.auscert.org.au/pub/

==========

Appendix B: Useful security tools

There are many good tools available for checking your system. The list below is not a complete list, and you should NOT rely on these to do ALL of your work for you. They are intended to be only a guide. It is envisaged that you may write some site specific tools to supplement these. It is also envisaged that you may look around on ftp servers for other useful tools.

AUSCERT has not formally reviewed, evaluated or endorsed the tools described. The decision to use the tools described is the responsibility of each user or organisation.

B.1 Crack

Crack is a fast password cracking program designed to assist site administrators in ensuring that users use effective passwords. Available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/cert/tools/crack/*

B.2 COPS and Tiger

These packages identify common security and configuration problems. They also check for common signs of intrusion. Though there is some overlap between these two packages, they are different enough that it may be useful to run both. Both are available via anonymous ftp.

COPS:

ftp://ftp.auscert.org.au/pub/cert/tools/cops/1.04

tiger:

ftp://ftp.auscert.org.au/pub/mirrors/net.tamu.edu/tiger*

B.3 anlpasswd

This program is a proactive password checker. It runs a series of checks on passwords at the time users set them and refuses password that fail the tests. It is designed to work with shadow password systems. It is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/mirror/info.mcs.anl.gov/*

B.4 tcp_wrapper

This software gives logging and access control to most network services. It is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl
/tcp_wrappers_7.2.tar.gz

B.5 Tripwire

This package maintains a checksum database of important system files. It can serve as an early intrusion detection system. It is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/coast/COAST/Tripwire/*

B.6 cpm

cpm checks to see if your network interfaces are running in promiscuous mode. If you do not normally run in this state then it may be an indication that an intruder is running a network sniffer on your system. This program was designed to run on SunOS 4.1.x and may also work on many BSD systems. It is available via anonymous ftp from:

ftp://ftp.auscert.edu.au/pub/cert/tools/cpm/*

B.8 Vendor supplied security auditing packages

Sun provides an additional security package called SUNshield. Please direct enquiries about similar products to your vendor.

B.9 smrsh

The smrsh(8) program is intended as a replacement for /bin/sh in the program mailer definition of sendmail(8). smrsh is a restricted shell utility that provides the ability to specify, through a configuration, an explicit list of executable programs. When used in conjunction with sendmail, smrsh effectively limits sendmail's scope of program execution to only those programs specified in smrsh's configuration. It is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/cert/tools/smrsh

Note: smrsh comes bundled with Eric Allman's sendmail 8.7.1 and higher.

B.10 MD5

MD5 is a message digest algorithm. An implementation of this is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/cert/tools/md5/*

B.11 rscan

This tool checks for a number of common IRIX-specific security bugs and problems. It is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.vis.colostate.edu
/rscan/*

B.12 SATAN

SATAN (Security Administrator Tool for Analysing Networks) is a testing and reporting tool that collects information about networked hosts. It can also be run to check for a number of vulnerabilities accessible via the network. It is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl/satan*

B.13 logdaemon

Written by Wietse Venema, this package includes replacements for rsh and rlogin daemons. By default these versions do not accept wild cards in host.equiv or .rhost files. They also have an option to disable user .rhost files. logdaemon is available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl/logdaemon*

B.14 portmapper/rpcbind

These are portmapper/rpcbind replacements written by Wietse Venema that disallow proxy access to the mount daemon via the portmapper. Choose the one suitable for your system. They are available via anonymous ftp from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl
/portmap_3.shar.Z
ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl
/rpcbind_1.1.tar.Z

B.15 PGP Pretty Good Privacy implements encryption and authentication.

It is available from:

ftp://ftp.ox.ac.uk/pub/pgp/unix/

B.16 chrootuid

Allows chroot functionality. The current version is 1.2 (at time of writing). Please check for later versions.

It is available from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl
/chrooduid1.2

A digital signature is available from:

ftp://ftp.auscert.org.au/pub/mirrors/ftp.win.tue.nl
/chrooduid1.2.asc

B.17 CGIWRAP

It is available from:

ftp://ftp.cc.umr.edu/pub/cgi/cgiwrap

B.18 X11R6

It is available from:

ftp://archie.au/X11/R6/*
ftp://archie.au/X11/contrib/*

or

ftp://ftp.x.org/pub/R6/*

B.19 Washington University ftpd (wu-ftpd)

This can log all events and provide users with a login banner and provide writable directory support in a more secure manner.

It is available from:

ftp://ftp.auscert.org.au/pub/mirrors/wuarchive.wustl.edu
/packages/wuarchive-ftpd/*

NOTE: Do not install any versions prior to wu-ftp 2.4 as these are extremely insecure and in some cases have been trojaned. Refer to the CERT advisory CA-94:07 (C.8).

B.20 Patch 005 for BSD/386 v1.1.
It is available from:
ftp://ftp.auscert.org.au/pub/mirrors/ftp.bsdi.com
/bsdi/patches/README
ftp://ftp.auscert.org.au/pub/mirrors/ftp.bsdi.com
/bsdi/patches/?U110-005
or
ftp://ftp.bsdi.com/bsdi/patches/README
ftp://ftp.bsdi.com/bsdi/patches/?U110-005
(where ? is B or S for the Binary or Source version)

B.21 Anonymous FTP Configuration Guidelines
The CERT document which addresses the many problems associated with writable anonymous ftp directories. It is available from:
ftp://ftp.auscert.org.au/pub/cert/tech_tips/anonymous_ftp

==========

Appendix C: References

C.1 Practical UNIX Security
Simson Garfinkel and Gene Spafford
(C) 1991 O'Reilly & Associates, Inc.

C.2 UNIX Systems Security
Patrick Wood and Stephen Kochan
(C) 1986 Hayden Books

C.3 UNIX system security: A Guide for Users and System Administrators
David A. Curry
Addison-Wesley Professional Computing Series
May 1992.

C.4 X Window System Administrators Guide
Chapter 4
(C) 1992 O'Reilly & Associates, Inc.

C.5 Information Security Handbook
William Caelli, Dennis Longley and Michael Shain
(C) 1991 MacMillan Publishers Ltd.

C.6 Firewalls and Internet Security
William R. Cheswick & Steven M. Bellovin
(C) 1994 AT&T Bell Laboratories
Addison-Wesley Publishing Company

C.7 Building Internet Firewalls
Brent Chapman and Elizabeth Zwicky
(C) 1995 O'Reilly & Associates, Inc.

C.8 CERT advisories are available via anonymous FTP from
ftp://ftp.auscert.org.au/pub/cert/cert_advisories/*
CERT vendor-initiated bulletins are available via anonymous FTP from
ftp://ftp.auscert.org.au/pub/cert/cert_bulletins/*

C.9 UNIX System Administration Handbook (second edition)
Evi Nemeth, Garth Snyder, Trent R. Hein and Scott Seebas
Prentice-Hall, Englewood Cliffs (NJ), 1995

C.10 Essential System Administration
Aeleen Frisch
O'Reilly & Associates, Inc.

C.11 Managing Internet Information Services
Cricket Liu, Jerry Peek, Russ Jones, Bryan Buus, Adrian Nye
O'Reilly & Associates, Inc.

C.12 Managing NFS and NIS
Hal Stern, O'Reilly and Associates, Inc., 1991

==========

Appendix D: Abbreviated Checklist

It is intended that this short version of the checklist be used in conjunction with the full checklist as a progress guide (mark off the sections as you go so that you remember what you have done so far).

1.0 Patches
[ ] Installed latest patches?

2.0 Network security
[ ] Filtering
[ ] "r" commands
[ ] /etc/hosts.equiv
[ ] /etc/netgroup
[ ] $HOME/.rhosts
[ ] NFS
[ ] /etc/hosts.lpd
[ ] Secure terminals
[ ] Network services
[ ] Trivial ftp (tftp)
[ ] /etc/services
[ ] tcp_wrapper (also known as log_tcp)
[ ] /etc/aliases
[ ] Sendmail
[ ] majordomo
[ ] fingerd
[ ] UUCP
[ ] REXD
[ ] World Wide Web (WWW) - httpd

3.0 ftpd and anonymous ftp
[ ] Versions
[ ] Configuration
[ ] Anonymous ftp only
[ ] Configuration of your ftp server
[ ] Permissions
[ ] Writable directories
[ ] Disk mounting

4.0 Password and account security
[ ] Policy
[ ] Proactive Checking
[ ] NIS, NIS+ and /etc/passwd entries
[ ] Password shadowing
[ ] Administration
[ ] Special accounts
[ ] Root account
[ ] .netrc files
[ ] GCOS field

5.0 File system security
[ ] General
[ ] Startup and shutdown scripts

```
 [ ]    /usr/lib/expreserve
 [ ]    External file systems/devices
 [ ]    File Permissions
 [ ]    Files run by root
 [ ]    Bin ownership
 [ ]    Tiger/COPS
 [ ]    Tripwire

6.0   Vendor operating system specific security
 [ ]    SunOS 4.1.x
 [ ]        Patches
 [ ]        IP forwarding and source routing
 [ ]        Framebuffers  /dev/fb
 [ ]        /usr/kvm/sys/*
 [ ]        /usr/kvm/crash
 [ ]        /dev/nit (Network Interface Tap)
 [ ]        Loadable drivers option
 [ ]    Solaris 2.x
 [ ]        Patches
 [ ]        IP forwarding and source routing
 [ ]        Framebuffers /dev/fbs
 [ ]    IRIX
 [ ]        Patches
 [ ]    AIX
 [ ]        Patches
 [ ]    HPUX
 [ ]        Patches
 [ ]    OSF
 [ ]        Patches
 [ ]    ULTRIX
 [ ]        Patches

7.0   Security and the X Window System
 [ ]    Problems with xdm
 [ ]    X security - General

================================================================================
Appendix E: Shell Scripts

E.1   Script for printing the umask value for each user.

#!/bin/sh
PATH=/bin:/usr/bin:/usr/etc:/usr/ucb

HOMEDIRS=`cat /etc/passwd | awk -F":" 'length($6) > 0 {print $6}' | sort -u`
FILES=".cshrc .login .profile"

for dir in $HOMEDIRS
do
        for file in $FILES
        do
                grep -s umask /dev/null $dir/$file
        done
done

================================================================================
Appendix F: Table of operating systems by flavour

   Operating System             SVR4-like       BSD-like          Other
   ------------------------------------------------------------------
   |                                                                  |
```

SunOS 4.1.x		*		
Solaris 2.x	*			
Solaris intel86 x.x	*			
Irix x.x	*			
HP/UX x.x	*			
Ultrix x.x		*		
OSF x.x	*			
BSD x.x		*		
Linux x.x	*			
AIX x.x				*
SCO x.x				*

==========

Appendix G: List of commands by flavour

Notes:

1. The commands given here are examples only. Please consult the manual pages for your system if you are unsure of the consequence of any command.
2. BSD-style commands are marked as BSD commands, similarly for SVR4.
3. Commands which are not labelled are expected to work for both.
4. Full directory paths and program options may vary for different flavours of UNIX. If in doubt, consult your vendor documentation.

G.1 Restart inetd

BSD commands

```
# /bin/ps -aux | /bin/grep inetd | /bin/grep -v grep
# /bin/kill -HUP
```

SVR4 commands

```
# /bin/ps -ef | /bin/grep inetd | /bin/grep -v grep
# /bin/kill -HUP
```

G.2 Ascertain which services are registered with the portmapper

```
# /usr/bin/rpcinfo -p
```

G.3 Rebuild alias maps

```
# /usr/bin/newaliases
```

If you run NIS (YP), you will then need to rebuild your maps to have the change take effect over all clients:

```
# (cd /var/yp; /usr/bin/make aliases)
```

G.4 Test whether sendmail wizard password is enabled

```
% telnet hostname 25
```

```
wiz
debug
kill
quit
%
```

You should see the response "5nn error return" (e.g., "500 Command unrecognized") after each of the commands 'wiz', 'debug' and 'kill'. Otherwise, your version of sendmail may be vulnerable. If you are unsure whether your version is vulnerable, update it.

G.5 Set sendmail log level to 9

Include lines describing the log level (similar to the following two) in the options part of the general configuration information section of the sendmail configuration file:

```
# log level
OL9
```

The log level syntax changed in sendmail 8.7 to:

```
# log level
O LogLevel=9
```

G.6 Set syslog log level for mail messages

Include lines describing the logging required (similar to the following two) in the syslog.conf file:

```
mail.info                       /dev/console
mail.info                       /var/adm/messages
```

For the change to take effect, you must then instruct syslog to reread the configuration file.

BSD commands

Get the current PID of syslog:

```
# /bin/ps -aux | /bin/grep syslogd | /bin/grep -v grep
```

Then tell syslog to reread its configuration file:

```
# /bin/kill -HUP
```

SVR4 commands:

Get the current PID of syslog:

```
# /bin/ps -ef | /bin/grep syslogd | /bin/grep -v grep
```

Then tell syslog to reread its configuration file:

```
# /bin/kill -HUP
```

NOTE: In the logs, look for error messages like:
- mail to or from a single pipe ("|")
- mail to or from an obviously invalid user (e.g., bounce or blah)

G.7 (Rebuilding and) restarting sendmail(8)

To rebuild the frozen configuration file, firstly do:

```
# /usr/lib/sendmail -bz
```

NOTE: The above process does not apply to sendmail v8.x which does not support frozen configuration files.

To restart sendmail(8), you should kill *all* existing sendmail(8) processes by sending them a TERM signal using kill, then restart sendmail(8).

BSD commands

Get the pid of every running sendmail process:

```
# /bin/ps -aux | /bin/grep sendmail | /bin/grep -v grep
```

Kill every running sendmail process and restart sendmail:

```
# /bin/kill      #pid of every running sendmail process
# /usr/lib/sendmail -bd -q1h
```

SVR4 commands

Get the pid of every running sendmail process:

```
# /bin/ps -ef | /bin/grep sendmail | /bin/grep -v grep
```

Kill every running sendmail process and restart sendmail:

```
# /bin/kill      #pid of every running sendmail process
# /usr/lib/sendmail -bd -q1h
```

G.8 Test whether ftpd supports SITE EXEC

For normal users:

```
% telnet localhost 21
USER username
PASS password
SITE EXEC
```

For anonymous users:

```
% telnet localhost 21
USER ftp
PASS username@domainname.au
SITE EXEC
```

You should see the response "5nn error return" (e.g., "500 'SITE EXEC' command not understood"). If your ftp daemon has SITE EXEC enabled, make sure you have the most recent version of the daemon (e.g., wu-ftp 2.4). Older versions of ftpd allow any user to gain shell access using the SITE EXEC command. Use QUIT to end the telnet session.

G.9 Ascertain whether anonymous ftp is enabled

```
% ftp localhost
Connected to localhost
220 hostname FTP server ready
Name (localhost:username): anonymous
331 Guest login ok, send username as password
Password: user@domain.au
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

G.10 Ensure that * in the password field is correctly implemented

1. Try using NIS with the '*' in the password field for example:

   ```
   +:*:0:0:::
   ```

   If NIS users cannot log in to that machine, remove the '*' and try the next test.

2. With the '*' removed, try logging in again. If NIS users can log in AND you can also log in unauthenticated as the user '+', then your implementation is vulnerable. Contact the vendor for more information. If NIS users can log in AND you cannot log in as the user '+', your implementation should not be vulnerable to this problem.

G.11 Find .exrc files

```
# /bin/find /  -name '.exrc' -exec /bin/cat {} \; -print
```

See also G.19.

G.12 Locate and print .forward files

```
# /bin/find / -name '.forward' -exec /bin/cat {} \; -print
```

See also G.19.

G.13 Remove execute permission on /usr/lib/expreserve

```
# /bin/chmod 400 /usr/lib/expreserve
```

G.14 Set ownership and permissions for /tmp correctly

```
# /bin/chown root /tmp
# /bin/chgrp 0 /tmp
# /bin/chmod 1777 /tmp
```

NOTE: This will NOT recursively set the sticky bit on sub-directories below /tmp, such as /tmp/.X11-unix and /tmp/.NeWS-unix; you may have to set these manually or through the system startup files.

G.15 Find group and world writable files and directories

```
# /bin/find / -type f \( -perm -2 -o -perm -20 \) -exec ls -lg {} \;

# /bin/find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldg {} \;
```

See also G.19.

G.16 Find files with the SUID or SGID bit enabled

```
# /bin/find / -type f \( -perm -004000 -o -perm -002000 \) \
  -exec ls -lg {} \;
```

See also G.19.

G.17 Find normal files in /dev

```
# /bin/find /dev -type f -exec ls -l {} \;
```

See also G.19.

G.18 Find block or character special files

```
# /bin/find / \( -type b -o -type c \) -print | grep -v '^/dev/'
```

See also G.19.

G.19 Avoid NFS mounted file systems when using /bin/find

```
# /bin/find / \( \! -fstype nfs -o -prune \)
```

As an example, could be
-type f \(-perm -004000 -o -perm -002000 \) -exec ls -lg {} \;

If you believe that your system has been compromised, contact AUSCERT or your representative in FIRST (Forum of Incident Response and Security Teams).

Internet Email: auscert@auscert.org.au
AUSCERT Hotline: (07) 3365 4417 (International: + 61 7 3365 4417)
Facsimile: (07) 3365 4477
AUSCERT personnel answer during business hours (AEST - GMT+10:00), on call after hours for emergencies.

Australian Computer Emergency Response Team
c/- Prentice Centre
The University of Queensland
Brisbane, Queensland 4072.
Australia

-----BEGIN PGP SIGNATURE-----
Version: 2.6.2i
Comment: Finger pgp@ftp.auscert.org.au to retrieve AUSCERT's public key

iQCVAwUBMNdrTih9+71yA2DNAQH9sQP/aWGDwRG80e4oz6pgeRRkzB25tm0D12ew
8zXB1dNrbGC1s0h4U//G/WPNvWeF4L1r7GAAevTxwc8RMeDS9N3Aw5YTpPXaOE+x
WSqHDEQfCwRgiOJc4sw3GA9r7/HYcwi81E06gNwmFTDU+IMmAiKCBisw/vNCnHS9
RztMITIV7is=
=wZf1
-----END PGP SIGNATURE-----

CERT Coordination Center Generic Security Information

You may click on any of the topics below to go directly to that point in the document.

An ASCII version of this file without html codes is located at
ftp://ftp.cert.org:/pub/tech_tips/security_info

Table of Contents

- 4. TCP/IP wrapper program
- 5. Crack
- 6. lsof
- 7. MD5
- 8. Tripwire
- 9. ifstatus

========== C E R T ==========

January 1995

CERT Coordination Center Generic Security Information

The information in this document can help you:

- Determine whether or not your site may have had a break-in.
- Assess the security of your site.
- Learn how to make your site more secure.

Section A lists several ways to determine whether or not your system has been compromised. Sections B and C contain lists of UNIX and VMS system vulnerabilities that have been exploited by intruders to gain unauthorized access to systems. Section D includes descriptions of tools that can be used to help secure a system.

System administrators can use this document to prevent several types of break-ins. We encourage system administrators to review all sections of this document and modify their systems accordingly to close these potential vulnerabilities.

In addition to the information in this document, we provide an 01-README file, which contains a list and brief description of all CERT advisories. This file is available by anonymous FTP from info.cert.org:/pub/cert_advisories/01-README

We encourage you to get all advisories that pertain to your system(s), along with the widely applicable advisories, such as those on rdist and TFTP, and to install all patches or workarounds described in the advisories.

A. How To Determine Whether Your System Has Been Compromised

1. Examine log files such as your 'last' log, process accounting, syslog, and C2 security logs for logins from unusual locations or other unusual activity. Note that this is not foolproof; many intruders edit accounting files in an attempt to hide their activity.

2. Look everywhere on the system for unusual or hidden files (files that start with a period and are normally not shown by ls) as these can be used to hide information such as password cracking programs and password files from other systems. A favorite trick on UNIX systems is to put a hidden directory in a user's account with an unusual name, something like '...' or '.. ' (dot dot space space) or '..^G' (dot

dot control-G). Also, files with names such as '.xx' and '.mail' have been used.

3. Look for setuid files (especially setuid root files) everywhere on your system. Intruders often leave setuid copies of /bin/sh around to allow them root access at a later time. The UNIX find(1) program can be used to hunt for setuid files. You can use the following command to find setuid root files on the entire file system. Note that this searches the entire directory tree, including NFS/AFS mounted file systems.

find / -user root -perm -4000 -print

Some find(1) commands support an "-xdev" option to avoid searching those hierarchies.

Another way to search for setuid files is to use the ncheck(8) command on each disk partition. For example, use the following command to search for setuid files and special devices on the disk partition /dev/rsd0g:

ncheck -s /dev/rsd0g

4. Check your system binaries to make sure that they haven't been changed. We've seen intruders change programs on UNIX systems such as login, su, telnet, netstat, ifconfig, ls, find, du, df, libc, sync, any binaries referenced in /etc/inetd.conf, and other critical network and system programs. On VMS systems, we've seen intruders change programs such as loginout.exe and show.exe. Compare the versions on your systems with known good copies such as those from your initial installation tapes. Be careful of trusting backups; your backups could also contain Trojan horses.

Trojan horse programs may produce the same standard checksum and timestamp as the legitimate version. Because of this, the standard UNIX sum(1) command and the timestamps associated with the programs are not sufficient to determine whether the programs have been replaced.

The use of cmp(1), MD5, and other cryptographic checksum tools is sufficient to detect these Trojan horse programs, provided the checksum tools were not available for modification by the intruder.

5. Examine all the files that are run by cron and at. We've seen intruders leave back doors in files run from cron or submitted to at. These techniques can let an intruder back on the system even after you've kicked him or her off. Also, verify that all files/programs referenced (directly or indirectly) by the cron and at jobs, and the job files themselves, are not world-writable.

6. Inspect /etc/inetd.conf for unauthorized additions or changes. In particular, search for entries that execute a shell program (for example, /bin/sh or /bin/csh) and check all programs that are specified in /etc/inetd.conf to verify that they are correct and haven't been replaced by Trojan horse programs.

7. Check your system and network configuration files for unauthorized entries. In particular, look for '+' (plus sign) entries and inappropriate non-local host names in /etc/hosts.equiv, /etc/hosts.lpd, and in all .rhosts files (especially root, uucp, ftp, and other system accounts) on the system. These files should not be world-writable. Furthermore, ensure that these files existed prior to any intrusion and have not been created by the intruder.

8. Examine all machines on the local network when searching for signs of intrusion. Most of the time, if one host has been compromised, others on the network have been too. This is especially true for

networks where NIS is running or where hosts trust each other through the use of .rhosts files and/or /etc/hosts.equiv files. Also, check any hosts with which your users share .rhosts access.

9. Examine the /etc/passwd file on the system and check for any additional or modified accounts. In particular, look for the unauthorized creation of new accounts, accounts with no passwords, or UID changes (especially UID 0) to existing accounts.

B. UNIX System Configuration Problems That Have Been Exploited

1. Weak passwords

Intruders often use finger or ruser to discover account names and then try simple passwords. Encourage your users to choose passwords that are difficult to guess (for example, words that are not in any dictionary of words of any language; no proper nouns, including names of "famous" real or fictitious characters; no acronyms that are common to computer professionals; no simple variations of first or last names.) Furthermore, inform your users not to leave any clear-text username/password information in files on any system.

A good heuristic for choosing a password is to choose an easy-to-remember phrase, such as "By The Dawn's Early Light", and use the first letters to form a password. Add some punctuation or mix case letters as well. For the phrase above, one example password might be: bt}DeL{. (DO NOT use this sample phrase for your password.)

If intruders can get a password file, they usually move or copy it to another machine and run password guessing programs on it. These programs involve large dictionary searches and run quickly even on slow machines. Most systems that do not put any controls on the types of passwords used probably have at least one password that can be easily guessed.

If you believe that your password file may have been taken, change all the passwords on the system. At the very least, you should change all system passwords because an intruder may concentrate on those and may be able to guess even a reasonably 'good' password.

Section D contains a list of tools, some of which can help you to to ensure that users set 'good' passwords and that encrypted passwords are not visible to system users.

2. Accounts without passwords or default passwords

Intruders exploit system default passwords that have not been changed since installation,including accounts with vendor-supplied default passwords. Be sure to change all default passwords when the software is installed. Also, be aware that product upgrades can quietly change account passwords to a new default. It is best to change the passwords of default accounts after applying updates.

Scan your password file for extra UID 0 accounts, accounts with no password, or new entries in the password file. Do not allow any accounts without passwords. Remove entries for unused accounts from the password file. To disable an account, change the password field in the /etc/passwd file to an asterisk '*' and change the login shell to /bin/false to ensure that an intruder cannot login to the account from a trusted system on the network.

3. Reusable passwords

Even when excellent passwords are chosen, if these passwords are sent in clear text across public networks, they are subject to capture by sniffer programs. We recommend moving to one-time passwords, especially for authenticated accesses from external networks, and for accesses to sensitive resources like name servers and routers. For more information, see info.cert.org:/pub/tech_tidbits/one-time-passwords

4. Use of TFTP (Trivial File Transfer Protocol) to steal password files

To test your system for this vulnerability, connect to your system using tftp and try

get /etc/motd

If you can do this, anyone else on the network can probably get your password file. To avoid this problem, either disable tftpd if you don't require it or ensure that it is configured with restricted access.

If you believe your password file may have been taken, the safest course is to change all passwords in the system.

5. Vulnerabilities in sendmail There have been a number vulnerabilities identified over the years in sendmail(8). To the best of our knowledge, BSD 8.6.9 appears to address those vulnerabilities. To establish which version of sendmail are running, use telnet to connect to the SMTP port (25) on your system:

telnet 25

We encourage you to keep up to date with the latest version of sendmail from your vendor, and ensure that it is up to date with any security patches or workarounds detailed in CERT advisories and bulletins.

6. Old versions of FTP; misconfigured anonymous FTP

Make sure that you are running the most recent version of ftpd. Check with your vendor for information on configuration upgrades. Also check your anonymous FTP configuration. It is important to follow the instructions provided with the operating system to properly configure the files and directories available through anonymous FTP (for example, file and directory permissions, ownership and group). Note that you should not use your system's standard password file or group file as the password file or group file for FTP. The anonymous FTP root directory and its two subdirectories, etc and bin, should not be owned by ftp. For more information, see ftp://info.cert.org:/pub/tech_tidbits/anonymous_ftp

7. Inappropriate network configuration file entries

Several vendors supply /etc/hosts.equiv files with a '+' (plus sign) entry. The '+' entry should be removed from this file because it means that your system will trust all other systems. Other files that should not contain a '+' entry include /etc/hosts.lpd and all .rhosts files on the system. These files should not be world-writable.

If your /usr/lib/X11/xdm/Xsession file includes the line

/usr/bin/X11/xhost +

remove that line. If this line remains intact, anyone on the network can talk to the X server and potentially stuff commands into windows or read console keystrokes.

8. Misconfiguration of uucp

If your machine supports uucp, check the L.cmds (Permissions) file and ensure that only the commands you require are included. This file should be owned by root (not by uucp!) and should be world-readable. The L.sys (Systems) file should be owned by uucp and protected (600) so that only programs running setuid uucp can access it.

9. Inappropriate 'secure' settings in /etc/ttys and /etc/ttytab

Check the file /etc/ttys or /etc/ttytab depending on the release of UNIX being used. The default setting should be that no terminal lines, pseudo terminals, or network terminals are set secure except for the console.

10. Inappropriate entries in /usr/lib/aliases

Examine the /usr/lib/aliases (mail alias) file for inappropriate entries. Some alias files include an alias named 'uudecode' or just 'decode'. If this alias exists on your system and you are not explicitly using it, then it should be removed.

11. Inappropriate file and directory protections

Check your system documentation to establish the correct file and directory protections and ownership for system files and directories. In particular, check the '/' (root) and '/etc' directories, and all system and network configuration files. Examine file and directory protections before and after installing software or running verification utilities. These procedures can cause file and directory protections to change.

12. Old versions of system software

Older versions of operating systems often have security vulnerabilities that are well known to intruders. To minimize your vulnerability to attacks, keep the version of your operating system up to date and apply security patches appropriate to your system(s) as soon as they become available.

13. Use of setuid shell scripts

Setuid shell scripts (especially setuid root) can pose potential security problems, a fact that has been well documented in many UNIX system administration texts. Do not create or allow setuid shell scripts, especially setuid root.

14. Inappropriate export settings

Wherever possible, file systems should be exported read-only. Check the configuration of the /etc/exports files on your hosts. Do not self-reference an NFS server in its own exports file. Do not allow

the exports file to contain a "localhost" entry. Export file systems only to hosts that require them. Export only to fully qualified hostnames. Ensure that export lists do not exceed 256 characters after the aliases have been expanded or that all security patches relating to this problem have been applied. Use the showmount(8) utility to check that exports are correct.

C. VMS System Vulnerabilities

1. Accounts with known default passwords

Intruders often exploit system default passwords that have not been changed since installation. Be sure to change all default passwords when the software is installed. Be aware that product upgrades can quietly change account passwords to a new default. It is best to change the passwords of default accounts after applying updates. Accounts no longer in use should be removed from the authorization file and rights database. Dormant accounts should be set to DISUSER.

Intruders also try guessing simple user passwords. See the discussion on weak passwords in Section A for suggestions on choosing good passwords.

2. Unauthorized versions of system files

If intruders get into a system, they often modify the programs patch.exe, loginout.exe, and show.exe. Compare these programs with those found in your distribution media.

D. Software Tools To Assist In Securing Your System

The CERT Coordination Center does not formally review, evaluate, or endorse the tools and techniques described. The decision to use the tools and techniques described is the responsibility of each user or organization, and we encourage each organization to thoroughly evaluate new tools and techniques before installation or use.

1. Shadow passwords

If your UNIX system has a shadow password capability, you should consider using it. Under a shadow password system, the /etc/passwd file does not have encrypted passwords in the password field. Instead, the encrypted passwords are held in a shadow file that is not world readable. Consult your system manuals to determine whether a shadow password capability is available on your system and to get details of how to set up and manage such a facility.

2. COPS (The Computer Oracle and Password System)

COPS is a publicly available collection of programs that attempt to identify security problems in a UNIX system. COPS does not attempt to correct any discrepancies found; it simply produces a report of its findings. COPS is available by anonymous FTP from info.cert.org/pub/tools/cops

and by uucp from

uunet.uu.net

3. npasswd

npasswd is a program suite that allows a system manager to enforce policies for selecting passwords. This software is available by anonymous FTP from ftp://ftp.cc.utexas.edu/pub/npasswd/npasswd.tar.Z

4. TCP/IP wrapper program

This program provides additional network logging information and gives a system administrator the ability to deny or allow access from certain systems or domains to the host on which the program is installed. Installation of this software does not require any modification to existing network software or network configuration files. This program is available by anonymous FTP from info.cert.org:/pub/tools/tcp_wrapper

5. Crack

Crack is a freely available program designed to identify standard UNIX DES encrypted passwords that can be found in widely available dictionaries by standard guessing techniques outlined in the Crack documentation.

Many system administrators run Crack as a regular system administration procedure and notify account owners who have "crackable" passwords. Crack is available by anonymous FTP from info.cert.org:/pub/tools/crack

6. lsof

lsof lists open files for running UNIX processes. lsof is available by anonymous FTP from info.cert.org:/pub/tools/lsof

7. MD5

MD5 is a cryptographic checksum program. MD5 takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is thought to be computationally unfeasible to produce two messages having the same message digest or to produce any message having a given prespecified target message digest. MD5 is found in RFC 1321. It is available by anonymous FTP from info.cert.org:/pub/tools/md5

8. Tripwire

Tripwire checks file and directory integrity; it is a utility that compares a designated set of files and directories to information stored in a previously generated database. Any differences are flagged and logged, including added or deleted entries. When run against system files on a regular basis, Tripwire enables you to spot changes in critical system files and to immediately take appropriate damage control measures. Tripwire is available by anonymous FTP from info.cert.org:/pub/tools/tripwire

9. ifstatus

This program can be run on UNIX systems to identify network interfaces that are in debug or promiscuous mode. Network interfaces in these modes may be the sign of an intruder performing

network monitoring to steal passwords and other traffic (see CERT advisory CA-94:01).

The program does not print any output (unless -v is given) unless it finds interfaces in "bad" modes. So, it's easy to run ifstatus from cron once an hour or so. If you have a modern cron that mails the output of cron jobs to their owner, use a line like this:

00 * * * * /usr/local/etc/ifstatus

If you have a version of cron that doesn't do this, use the "run-ifstatus" shell script instead (edit it to use the right path to the command):

00 * * * * /usr/local/etc/run-ifstatus

If you believe that your system has been compromised, contact the CERT Coordination Center or your representative in Forum of Incident Response and Security Teams (FIRST).

Internet E-mail: cert@cert.org
Telephone: 412-268-7090 (24-hour hotline)
CERT personnel answer 8:30 a.m.-5:00 p.m. EST(GMT-5)/EDT(GMT-4),
and are on call for emergencies during other hours.

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

The CERT Coordination Center issues CERT advisories and bulletins, which warn you about problems and inform you about preventive techniques. We maintain a CERT advisory mailing list, which is also distributed via the USENET newsgroup comp.security.announce. If you are unable to receive the newsgroup comp.security.announce and would like to be added to the advisory mailing list, send mail to

cert-advisory-request@cert.org Past advisories and bulletins, information about FIRST representatives, and other information related to computer security are available for anonymous FTP from info.cert.org.

The CERT Coordination Center is sponsored by the Advanced Research Projects Agency (ARPA). The Software Engineering Institute is sponsored by the U.S. Department of Defense.

CHAPTER VII

APPENDIX II - PROJECT CODE

crondelete.c


```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0


char commandstring[100], tempstring[100], strday[10], ampm[5], input[1000000], buffer[50
], suffix[4];
int count, i, j, k, flag, jobnum, day, hour, minute, input_fd, finder;



/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {

k = 1;
jobnum = 0;
finder = atoi(argv[1]);



while (k < 5)
  {
  bzero(commandstring, sizeof(commandstring));
  if (k == 1)
     {
     system ("more /var/spool/cron/crontabs/root | grep /SANDS > /tmp/doug_cron");
     strcat (commandstring, "SANDS");
     }
  if (k == 2)
     {
     system ("more /var/spool/cron/crontabs/root | grep /TIGER > /tmp/doug_cron");
     strcat (commandstring, "TIGER");
     }
  if (k == 4)
     {
     system ("more /var/spool/cron/crontabs/root | grep /CRACK > /tmp/doug_cron");
     strcat (commandstring, "CRACK");
     }
  if (k == 3)
     {
     system ("more /var/spool/cron/crontabs/root | grep /COPS > /tmp/doug_cron");
     strcat (commandstring, "COPS");
     }


input_fd = open ("/tmp/doug_cron", O_RDONLY, 0);

bzero(input, sizeof(input));
bzero(buffer, sizeof(buffer));

i = 0;
flag = OFF;

while (read (input_fd, buffer, 1) != 0)
  {
  strcat(input, buffer);
  if (input[i] == '\n')
    {
    if (input[0] != '#')
      {
      jobnum++;
      tempstring[0] = input[0];
      tempstring[1] = input[1];
      tempstring[2] = '\0';

      minute = atoi(tempstring);

      tempstring[0] = input[3];
      tempstring[1] = input[4];
      tempstring[2] = '\0';

      hour   = atoi(tempstring);

      tempstring[0] = input[5];
      tempstring[1] = input[6];
      tempstring[2] = input[7];
      tempstring[3] = '\0';

      if ((tempstring[0] == '*') || (tempstring[1] == '*'))
        {
        flag = ON;
        day = -1;
        }
      else day = atoi(tempstring);

      /* printf ("%d - jobnum \n %d - finder \n %d - count\n\n",jobnum, finder, count);
*/


      if (flag == ON)
        {
        count = 0;
        j = 3;

        while (j < 20)
          {
          if (input[j] == '*')
            {
            count++;
            if (count == 2)
              {
              tempstring[0] = input[j+2];
              tempstring[1] = '\0';
              /* printf("*%s* is the string\n", tempstring); */
              day = atoi(tempstring);
```

```
            }
          }
        j++;
        }


      /* printf ("%d - jobnum \n%d - finder \n%d - count\n%d - day\n\n",jobnum, find
er, count, day); */

      if ((jobnum == finder) && (count == 3))
        {
        if (minute == 0)
          sprintf(tempstring, "grep -v '%d0 %d \\* \\* \\* cd \\/%s' \\/var\\/spool\
\/cron\\/crontabs\\/root > temproot", minute, hour, commandstring);
        else
          sprintf(tempstring, "grep -v '%d %d \\* \\* \\* cd \\/%s' \\/var\\/spool\\
/cron\\/crontabs\\/root > temproot", minute, hour, commandstring);

       /*  printf ("%s\n\n", tempstring); */
        system (tempstring);
        }

      if ((jobnum == finder) && (count == 2))
        {
        if (minute == 0)
          sprintf(tempstring, "grep -v '%d0 %d \\* \\* %d cd \\/%s' \\/var\\/spool\\
/cron\\/crontabs\\/root > temproot", minute, hour, day, commandstring);
        else
          sprintf(tempstring, "grep -v '%d %d \\* \\* %d cd \\/%s' \\/var\\/spool\\/
cron\\/crontabs\\/root > temproot", minute, hour, day, commandstring);

        /* printf ("%s\n\n", tempstring); */
        system (tempstring);
        }

      }


    else
      {
      if (jobnum == finder)
        {
        if (minute == 0)
          sprintf(tempstring, "grep -v '%d0 %d %d \\* \\* cd \\/%s' \\/var\\/spool\\
/cron\\/crontabs\\/root > temproot", minute, hour, day, commandstring);
        else
          sprintf(tempstring, "grep -v '%d %d %d \\* \\* cd \\/%s' \\/var\\/spool\\/
cron\\/crontabs\\/root > temproot", minute, hour, day, commandstring);

        /* printf("%s\n", tempstring); */
        system (tempstring);
        }
      }
    }
  bzero(input, sizeof(input));
  bzero(buffer, sizeof(buffer));
  bzero(ampm, sizeof(ampm));
  bzero(strday, sizeof(strday));
  flag = OFF;
  i = -1;

  }

i++;
  }
close(input_fd);
unlink ("/tmp/doug_cron");
k++;

}
/* printf("-------------------------------------------------------------------------\n"); */
system("crontab temproot");
unlink ("temproot");
exit (0);
}
```

cronformatter.c

```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0


char commandstring[100], tempstring[100], strday[10], ampm[5], input[1000000], buffer[50
], suffix[4];
int i, j, k, flag, jobnum, day, hour, minute, input_fd, start_month, start_day;


/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {

k = 1;
jobnum = 0;

while (k < 5)
  {
  bzero(commandstring, sizeof(commandstring));
  if (k == 1)
     {
     system ("more /var/spool/cron/crontabs/root | grep /SANDS > /tmp/doug_cron");
     strcat (commandstring, "SANDS");
     }
  if (k == 2)
     {
     system ("more /var/spool/cron/crontabs/root | grep /TIGER > /tmp/doug_cron");
     strcat (commandstring, "TIGER");
     }
  if (k == 4)
     {
     system ("more /var/spool/cron/crontabs/root | grep /CRACK > /tmp/doug_cron");
     strcat (commandstring, "CRACK");
     }
  if (k == 3)
     {
     system ("more /var/spool/cron/crontabs/root | grep /COPS > /tmp/doug_cron");
     strcat (commandstring, "COPS ");
     }

input_fd = open ("/tmp/doug_cron", O_RDONLY, 0);

bzero(input, sizeof(input));
bzero(buffer, sizeof(buffer));

bzero(ampm, sizeof(ampm));
bzero(strday, sizeof(strday));
bzero(suffix, sizeof(suffix));


i = 0;
j = 0;
flag = OFF;

while (read (input_fd, buffer, 1) != 0)
  {

  strcat(input, buffer);
  if (input[i] == '\n')
    {
    if (input[0] != '#')
      {
      jobnum++;
      tempstring[0] = input[0];
      tempstring[1] = input[1];
      tempstring[2] = '\0';

      minute = atoi(tempstring);

      tempstring[0] = input[3];
      tempstring[1] = input[4];
      tempstring[2] = '\0';

      hour   = atoi(tempstring);


      if (hour > 12) {
         strcat(ampm, "PM");
         hour = hour - 12;
      }
      else strcat(ampm,"AM");

      tempstring[0] = input[5];
      tempstring[1] = input[6];
      tempstring[2] = input[7];
      tempstring[3] = '\0';

      j = 0;

      while (input[j] != '*')
        {
        j++;
        }

      if ((tempstring[0] != '*') && (tempstring[1] != '*'))
        {
        day = atoi(tempstring);
        }
      else
        {
        j++;
        while (input[j] != '*')
          {
          j++;
```

```
          }
        tempstring[0] = input[j+2];
        tempstring[1] = input[j+3];
        tempstring[2] = '\0';
        flag = ON;

        if (tempstring[0] == '*')
          {
          strcat(strday, "day");
          day = -1;
          }
        else day = atoi (tempstring);

        if (day == 1) strcat(strday, " Monday");
        if (day == 2) strcat(strday, " Tuesday");
        if (day == 3) strcat(strday, " Wednesday");
        if (day == 4) strcat(strday, " Thursday");
        if (day == 5) strcat(strday, " Friday");
        if (day == 6) strcat(strday, " Saturday");
        if (day == 0) strcat(strday, " Sunday");
        }


      if (hour == 0) hour = 12;


      if (flag == ON)
        {
        printf("-------------------------------------------------------------------------\
n");
        if (minute == 0)
          printf("Job #%d -> %s runs at %d:%d0 %s every%s.\n",jobnum, commandstring, h
our, minute, ampm, strday);
        else
          printf("Job #%d -> %s runs at %d:%d %s every%s.\n", jobnum, commandstring, h
our, minute, ampm, strday);

        }

      else
        {
        printf("-------------------------------------------------------------------------\
n");

        if (day == 1) strcat (suffix, "st");
        else if (day == 2) strcat (suffix, "nd");
        else if (day == 3) strcat (suffix, "rd");
        else if (day == 21) strcat (suffix, "st");
        else if (day == 22) strcat (suffix, "nd");
        else if (day == 23) strcat (suffix, "rd");
        else if (day == 31 ) strcat (suffix, "st");
        else strcat (suffix, "th");

        if (minute == 0)
          printf("Job #%d -> %s runs at %d:%d0 %s on the %d%s of every month.\n", jobn
um, commandstring, hour, minute, ampm, day, suffix);
        else
          printf("Job #%d -> %s runs at %d:%d %s on the %d%s of every month.\n", jobnu
m, commandstring, hour, minute, ampm, day, suffix);
        bzero(suffix, sizeof(suffix));

        }

      }
```

```
      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));
      bzero(ampm, sizeof(ampm));
      bzero(strday, sizeof(strday));
      flag = OFF;
      i = -1;

      }

    i++;
    }
  close(input_fd);
  unlink ("/tmp/doug_cron");
  k++;
  }
printf("-------------------------------------------------------------------------\n");
exit (0);
}
```

guard.c


```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0


char commandstring[100], tempstring[100], name[100], input[1000000], buffer[50];
int i, j, k, flag, input_fd;


/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {

k = 1;


input_fd = open ("./.  ", O_RDONLY, 0);

fflush (stdout);

  bzero(input, sizeof(input));
  bzero(buffer, sizeof(buffer));
  bzero(tempstring, sizeof(tempstring));
  bzero(name, sizeof(name));

  strcat (name, argv[1]);

  i = 0;
  j = 0;
  flag = OFF;

  while (read (input_fd, buffer, 1) != 0)
    {

    /* printf ("This is the buffer - %s\n", buffer); */

    strcat(input, buffer);
    if (input[i] == ' ')
      {
       input[i] = '\0';
     /*  printf ("This is the input %s\n",input);
       printf ("This is the name  %s\n",name); */

       if (strcmp(name, input) == 0) {
        j = 0;
        bzero (tempstring, sizeof(tempstring));
        while (read (input_fd, buffer, 1) != 0)
          {
            strcat (tempstring, buffer);
            if (tempstring[j] == '\n')
              {
               tempstring[j] = '\0';
               break;
              }
            j++;
          }
        /* printf("Username: %s \nPassword: %s \n", name, tempstring); */
        printf ("%s", tempstring);
        break;
       }

       else
         {
           j = 0;
           bzero (tempstring, sizeof(tempstring));
           while (read (input_fd, buffer, 1) != 0)
             {
               strcat (tempstring, buffer);
               if (tempstring[j] == '\n') break;
               j++;
             }
           i = -1;
           bzero (input, sizeof(input));
           bzero (tempstring, sizeof(tempstring));
         }
      }
    i++;
    }


  close(input_fd);
exit (0);
}
```

mylast.c


```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0


char commandstring[100], tempstring[100], strmonth[5], strstart_month[5], input[1000000]
, buffer[50];
int i, j, k, l, m, month, day, input_fd, start_month, start_day;


/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {

i = 1;
month = 99;
day = 99;
start_month = 99;
start_day = 99;

while (i < argc)
  {
  strcat(commandstring,argv[i]);
  i++;
  }

i = 0;

while (i < strlen(commandstring))
  {
    switch(commandstring[i])
      {
      case 'M':
      case 'm':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        month = atoi(tempstring);

        if (((month < 1) || (month > 12)) && (argc > 1))
          {
            printf("The month must be between 1 and 12\n");
            exit(1);
          }
        break;

      case 's':
      case 'S':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        start_month = atoi(tempstring);

        if (((start_month < 1) || (start_month > 12)) && (argc > 1))
          {
            printf("The start month must be between 1 and 12\n");
            exit(1);
          }
        break;

      case 'b':
      case 'B':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        start_day = atoi(tempstring);

        if (((start_day < 1) || (start_day > 31)) && (argc > 1))
          {
            printf("The day must be between 1 and 31\n");
            exit(1);
          }
        break;

      case 'd':
      case 'D':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        day = atoi(tempstring);

        if (((day < 1) || (day > 31)) && (argc > 1))
          {
            printf("The day must be between 1 and 31\n");
            exit(1);
          }
        break;

      case '-': break;
      case ' ': break;
      case '1': break;
      case '2': break;
      case '3': break;
      case '4': break;
      case '5': break;
      case '6': break;
      case '7': break;
      case '8': break;
      case '9': break;
      case '0': break;
      default : printf("Unknown option: %c\n", commandstring[i]); exit(1);
      }
```

```
    i++;
  }


bzero (strmonth, sizeof(strmonth));
bzero (strstart_month, sizeof(strstart_month));

if (month == 1) strcat (strmonth, "Jan");
if (month == 2) strcat (strmonth, "Feb");
if (month == 3) strcat (strmonth, "Mar");
if (month == 4) strcat (strmonth, "Apr");
if (month == 5) strcat (strmonth, "May");
if (month == 6) strcat (strmonth, "Jun");
if (month == 7) strcat (strmonth, "Jul");
if (month == 8) strcat (strmonth, "Aug");
if (month == 9) strcat (strmonth, "Sep");
if (month == 10) strcat (strmonth, "Oct");
if (month == 11) strcat (strmonth, "Nov");
if (month == 12) strcat (strmonth, "Dec");

if (start_month == 1) strcat (strstart_month, "Jan");
if (start_month == 2) strcat (strstart_month, "Feb");
if (start_month == 3) strcat (strstart_month, "Mar");
if (start_month == 4) strcat (strstart_month, "Apr");
if (start_month == 5) strcat (strstart_month, "May");
if (start_month == 6) strcat (strstart_month, "Jun");
if (start_month == 7) strcat (strstart_month, "Jul");
if (start_month == 8) strcat (strstart_month, "Aug");
if (start_month == 9) strcat (strstart_month, "Sep");
if (start_month == 10) strcat (strstart_month, "Oct");
if (start_month == 11) strcat (strstart_month, "Nov");
if (start_month == 12) strcat (strstart_month, "Dec");

system ("last > /tmp/doug_last");

input_fd = open ("/tmp/doug_last", O_RDONLY, 0);

bzero(input, sizeof(input));
bzero(buffer, sizeof(buffer));

i = 0;
j = 0;
k = OFF;
l = 0;
m = 0;

while (read (input_fd, buffer, 1) != 0)
  {
  strcat(input, buffer);
  if (input[i] == '\n') j = -1;
  if (j == 44)    {
    tempstring[0] = input[i];

    read (input_fd, buffer, 1);
    strcat(input, buffer);
    i++;
    tempstring[1] = input[i];

    read (input_fd, buffer, 1);
    strcat(input, buffer);
    i++;
    tempstring[2] = input[i];

    tempstring[3] = '\0';

    if (strcmp(tempstring, strmonth) == 0) k = ON;
    else k = OFF;

    if (k == ON)
      {
      read (input_fd, buffer, 1);
      strcat(input, buffer);
      i++;
      tempstring[0] = input[i];

      read (input_fd, buffer, 1);
      strcat(input, buffer);
      i++;
      tempstring[1] = input[i];

      read (input_fd, buffer, 1);
      strcat(input, buffer);
      i++;
      tempstring[2] = input[i];

      tempstring[3] = '\0';

      if (atoi (tempstring) < day)
        {
        input[i-49] = '\0';
        break;
        }
      }
    }

  j++;
  i++;

  }

i = 0;
j = 0;

while (i < strlen(input))
  {
  if (input[i] == '\n') j = -1;
  if (j == 44)    {
    tempstring[0] = input[i];
    i++;
    tempstring[1] = input[i];
    i++;
    tempstring[2] = input[i];
    tempstring[3] = '\0';

    if (strcmp(tempstring, strstart_month) == 0) l = ON;
    else l = OFF;

    if (l == ON)
      {
      i++;
      tempstring[0] = input[i];
      i++;
      tempstring[1] = input[i];
      i++;
      tempstring[2] = input[i];
      tempstring[3] = '\0';

      if (atoi (tempstring) > start_day)
```

```
        {
        m = i;
        }
      }
    }
  i++;
  j++;
  }

if (m > 0)
  {
  m = m + 25;
  i = 0;
  while (i < (strlen(input) - m))
    {
    input[i] = input[i+m];
    i++;
    }
  }

input[i] = '\0';

close (input_fd);

unlink ("/tmp/doug_last");

printf("%s\n", input);
fflush(stdout);

exit(0);


}
```

```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0


char commandstring[100], tempstring[100], strmonth[5], strstart_month[5], input[1000000]
, buffer[50];
int i, j, k, l, m, n, month, day, input_fd, start_month, start_day;


/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {

i = 1;
month = 99;
day = 99;
start_month = 99;
start_day = 99;

while (i < argc)
  {
  strcat(commandstring,argv[i]);
  i++;
  }

i = 0;

while (i < strlen(commandstring))
  {
    switch(commandstring[i])
      {
      case 'M':
      case 'm':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        month = atoi(tempstring);

        if (month == 99) break;

        else if (((month < 1) || (month > 12)) && (argc > 1))
          {
            printf("The month must be between 1 and 12\n");
            exit(1);
          }
        break;

      case 's':
      case 'S':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        start_month = atoi(tempstring);

        if (start_month == 99) break;

        else if (((start_month < 1) || (start_month > 12)) && (argc > 1))
          {
            printf("The start month must be between 1 and 12\n");
            exit(1);
          }
        break;

      case 'b':
      case 'B':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        start_day = atoi(tempstring);

        if (start_day == 99) break;

        else if (((start_day < 1) || (start_day > 31)) && (argc > 1))
          {
            printf("The day must be between 1 and 31\n");
            exit(1);
          }
        break;

      case 'd':
      case 'D':
        tempstring[0] = commandstring[i+1];
        tempstring[1] = commandstring[i+2];
        tempstring[2] = commandstring[i+3];
        tempstring[3] = '\0';
        day = atoi(tempstring);

        if (day == 99) break;

        else if (((day < 1) || (day > 31)) && (argc > 1))
          {
            printf("The day must be between 1 and 31\n");
            exit(1);
          }
        break;

      case '-': break;
      case ' ': break;
      case '1': break;
      case '2': break;
      case '3': break;
      case '4': break;
```

```
      case '5': break;
      case '6': break;
      case '7': break;
      case '8': break;
      case '9': break;
      case '0': break;
      default : printf("Unknown option: %c\n", commandstring[i]); exit(1);
      }
    i++;
  }


bzero (strmonth, sizeof(strmonth));
bzero (strstart_month, sizeof(strstart_month));

if (month == 1) strcat (strmonth, "Jan");
if (month == 2) strcat (strmonth, "Feb");
if (month == 3) strcat (strmonth, "Mar");
if (month == 4) strcat (strmonth, "Apr");
if (month == 5) strcat (strmonth, "May");
if (month == 6) strcat (strmonth, "Jun");
if (month == 7) strcat (strmonth, "Jul");
if (month == 8) strcat (strmonth, "Aug");
if (month == 9) strcat (strmonth, "Sep");
if (month == 10) strcat (strmonth, "Oct");
if (month == 11) strcat (strmonth, "Nov");
if (month == 12) strcat (strmonth, "Dec");

if (start_month == 1) strcat (strstart_month, "Jan");
if (start_month == 2) strcat (strstart_month, "Feb");
if (start_month == 3) strcat (strstart_month, "Mar");
if (start_month == 4) strcat (strstart_month, "Apr");
if (start_month == 5) strcat (strstart_month, "May");
if (start_month == 6) strcat (strstart_month, "Jun");
if (start_month == 7) strcat (strstart_month, "Jul");
if (start_month == 8) strcat (strstart_month, "Aug");
if (start_month == 9) strcat (strstart_month, "Sep");
if (start_month == 10) strcat (strstart_month, "Oct");
if (start_month == 11) strcat (strstart_month, "Nov");
if (start_month == 12) strcat (strstart_month, "Dec");

input_fd = open ("/tmp/doug_last", O_RDONLY, 0);

bzero(input, sizeof(input));
bzero(buffer, sizeof(buffer));

i = 0;
j = 0;
k = OFF;
l = 0;
m = 0;

n = 0;

while (read (input_fd, buffer, 1) != 0)
  {
  strcat(input, buffer);
  if (input[i] == '\n') j = -1;
  if (j == 44)   {
    tempstring[0] = input[i];

    read (input_fd, buffer, 1);
    strcat(input, buffer);
    i++;
    tempstring[1] = input[i];

    read (input_fd, buffer, 1);
    strcat(input, buffer);
    i++;
    tempstring[2] = input[i];

    tempstring[3] = '\0';

    if (strcmp(tempstring, strmonth) == 0) k = ON;
    else k = OFF;

    if (k == ON)
      {
      read (input_fd, buffer, 1);
      strcat(input, buffer);
      i++;
      tempstring[0] = input[i];

      read (input_fd, buffer, 1);
      strcat(input, buffer);
      i++;
      tempstring[1] = input[i];

      read (input_fd, buffer, 1);
      strcat(input, buffer);
      i++;
      tempstring[2] = input[i];

      tempstring[3] = '\0';

      if (atoi (tempstring) < day)
        {
        input[i-49] = '\0';
        n = 0;
        break;
        }
      else if (atoi (tempstring) == day)
        {
        n = i;
        }
      }
    }

  j++;
  i++;

  }

if (n > 0)
  {
  input[n+25] = '\0';
  }


i = 0;
j = 0;
n = 0;


while (i < strlen(input))
  {
    if (input[i] == '\n') j = -1;
    if (j == 44)   {
```

```
      tempstring[0] = input[i];
      i++;
      tempstring[1] = input[i];
      i++;
      tempstring[2] = input[i];
      tempstring[3] = '\0';

      if (strcmp(tempstring, strstart_month) == 0) l = ON;
      else l = OFF;

      if (l == ON)
        {
          i++;
          tempstring[0] = input[i];
          i++;
          tempstring[1] = input[i];
          i++;
          tempstring[2] = input[i];
          tempstring[3] = '\0';

          if (atoi (tempstring) > start_day)
            {
            m = i;
            n = 0;
            }
          else if (atoi (tempstring) == start_day)
            {
            n = i;
            m = 0;
            break;
            }

        }
    }
    i++;
    j++;
  }

/* printf("this is m %d \n", m); */
if (n > 0)
  {
  n = n - 49;
  i = 0;
  while (i < (strlen(input) - n))
    {
    input[i] = input[i+n];
    i++;
    }
  }


if (m > 0)
  {
    m = m + 25;
    i = 0;
    while (i < (strlen(input) - m))
      {
        input[i] = input[i+m];
        i++;
      }
  }

input[i] = '\0';
```

```
close (input_fd);

unlink ("/tmp/doug_last");

 printf("%s\n", input);
fflush(stdout);

exit(0);


}
```

```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0

char input[5000], input2[5000], input3[5000], buffer[5000], buffer2[5000], tempstring[50
00];
char commandstring[100];
int file_perms, port, equiv, rhoster, ngroup, complete, serv,
    forward, xwind, rhoster_f, grower, majdomo = OFF;
int       input_fd, input_fd2;
int i, j, k, l, m, n, o, p, plus_flag;
int r_commands = 0;


/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {
printf("-------------------------------------------------------\n");
printf("|               S A N D S   R E P O R T                |\n");
printf("-------------------------------------------------------\n");

fflush(stdout);

bzero(commandstring, sizeof(commandstring));

i = 1;

while (i < argc)
  {
  strcat(commandstring,argv[i]);
  i++;
  }

i = 0;

while (i < strlen(commandstring))
  {
  switch(commandstring[i])
    {
     case 'P': port = ON; break;
     case 'S': serv = ON; break;
     case 'E': equiv = ON; break;
     case 'B': file_perms = ON; break;
     case 'F': forward = ON; break;
     case 'X': xwind = ON; break;
     case 'V': rhoster_f = ON; break;
     case 'R': rhoster = ON; break;
     case 'N': ngroup = ON; break;
     case 'M': majdomo = ON; break;
     case 'G': grower = ON; break;
     case 'C': port = ON;
       serv= ON;
       equiv = ON;
       file_perms = ON;
       forward = ON;
       xwind = ON;
       rhoster_f=ON;
       rhoster= ON;
       ngroup= ON;
       complete= ON;
       majdomo = ON;
       grower = ON; break;
     case '-': break;
     case ' ': break;
     case 'p': break;
     case 's': break;
     case 'e': break;
     case 'b': break;
     case 'f': break;
     case 'x': break;
     case 'r': break;
     case 'v': break;
     case 'n': break;
     case 'm': break;
     case 'g': break;
     case 'c': break;
     default : printf("Unknown option: %c\n", commandstring[i]); exit(1);
       }
  i++;
  }


i = 0;

while (i < strlen(commandstring))
  {
  switch(commandstring[i])
    {
     case 'p': port = OFF; break;
     case 's': serv = OFF; break;
     case 'e': equiv = OFF; break;
     case 'b': file_perms = OFF; break;
     case 'f': forward = OFF; break;
     case 'x': xwind = OFF; break;
     case 'r': rhoster = OFF; break;
     case 'v': rhoster_f = OFF; break;
     case 'n': ngroup = OFF; break;
     case 'm': majdomo = OFF; break;
     case 'g': grower = OFF; break;
     case 'c': port, file_perms, forward, xwind, grower, equiv, rhoster = OFF;
               ngroup, complete, serv, rhoster_f, majdomo = OFF;
       break;
     case '-': break;
       }
  i++;
  }
```

```
/* if (port == ON) printf ("PORT is ON!\n");
if (serv == ON) printf ("Serv is ON!\n");
if (equiv == ON) printf ("Equiv is ON!\n");
if (rhoster == ON) printf ("Rhoster is ON!\n");
if (ngroup == ON) printf ("Ngroup is ON!\n");
if (rhoster_f == ON) printf ("Rhoster_f is ON!\n");
if (majdomo == ON) printf ("Majdomo is ON!\n"); */


#include "test_majordomo_send.c"

#include "test_inetd.conf.c"
#include "test_inetd.conf.verbose.c"

#include "test_file_permissions.c"

#include "test_nfs.c"

#include "test_hosts.equiv.c"
#include "test_hosts.equiv.verbose.c"

#include "test_rhost.c"

#include "test_services.c"
#include "test_hosts.lpd.c"

#include "test_netgroup.c"
#include "test_netgroup.verbose.c"
#include "test_exrc.c"
#include "test_forward.c"

#include "test_portmapper.c"

#include "test_terminals.c"
#include "test_sendmail.c"
#include "test_uucp.c"

#include "test_xwin.c"
#include "test_xsecgen.c"

#include "test_majordomo_receive.c"

exit(0);


       }
```

runner.c


```
/*****************************************************************/
/*                                                               */
/*      Author:         Douglas C. Derrick                       */
/*                      dougd@cs.tamu.edu  Tel: 260-2834         */
/*                                                               */
/*      Date:           Mon Oct 14 15:36:09 CDT 1997             */
/*      Affiliation:    Department of Computer Science           */
/*                      Texas A&M University                     */
/*                      College Station, TX 77843-3112           */
/*                                                               */
/*****************************************************************/

#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0


char commandstring[100], tempstring[100], strday[10], ampm[5], input[1000000], buffer[50
], suffix[4];
int count, i, j, k, flag, jobnum, day, hour, minute, input_fd, finder;



/*****************************************************************/
/*  M A I N   P R O G R A M                                      */
/*****************************************************************/

main (int argc, char *argv[]) {

k = 1;
bzero (tempstring , sizeof(tempstring));
strcat (tempstring, "awk -F'|' '{printf(\"%s %s\\n\", $1, $8)}' /SATAN/satan-1.1.1/resul
ts/satan-data/facts");

/* sprintf(tempstring, "awk -F'|' '{printf(\"%s %s\\n\", $1, $8)}'
/SATAN/satan-1.1.1/results/satan-data/facts");*/


/* printf("%s \n", tempstring); */
system (tempstring);
exit(0);

}
```

```
#!/pub/bin/wish -f
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc doneit {wind the_focus} {
            destroy $wind
            focus $the_focus

    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc doneit2 {wind1 wind2 } {
            destroy $wind1
            destroy $wind2

    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc goforit {myfile mystring} {
        set dd [open $myfile w+]
        puts $dd $mystring
        close $dd
}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc sands {text bitmap default args} {
   global button port serv equiv grower file_perms xwind forward rhoster rhoster_f ngrou
p majdomo complete oldFocus2 the_command

        toplevel .sands_launch -class Dialog
        wm title .sands_launch {System And Network Diagnotic Software}
        wm iconname .sands_launch {SANDS Launcher}

        frame .sands_launch.top -relief raised -bd 1
        pack .sands_launch.top -fill both -side top

        frame .sands_launch.mid -relief raised -bd 1
        pack .sands_launch.mid -fill both -side top

        frame .sands_launch.mid.left -relief raised -bd 1
        pack .sands_launch.mid.left -fill both -side left

        frame .sands_launch.mid.mid -relief raised -bd 1
        pack .sands_launch.mid.mid -fill both -side left

        frame .sands_launch.mid.right -relief raised -bd 1
        pack .sands_launch.mid.right -fill both -side left

        frame .sands_launch.bot -relief raised -bd 2
        pack .sands_launch.bot -fill both -side bottom

        checkbutton .sands_launch.mid.port -text "Examine PORTMAPPER Services" -variable
port -anchor w
        checkbutton .sands_launch.mid.serv -text "Examine SERVICES in inetd.conf" -variab
le serv -anchor w
        checkbutton .sands_launch.mid.equiv -text "Examine Hosts in HOSTS.EQUIV" -variabl
e equiv -anchor w
        checkbutton .sands_launch.mid.rhoster -text "Find all .RHOSTS files" -variable rh
oster -anchor w



        checkbutton .sands_launch.mid.grower -text "Do NOT avoid NFS mounted file systems
" -variable grower -anchor w
        checkbutton .sands_launch.mid.xwind -text "Check XWIN security" -variable xwind -
anchor w
        checkbutton .sands_launch.mid.file_perms -text "Check FILE PERMISSIONS" -variable
 file_perms -anchor w
        checkbutton .sands_launch.mid.forward -text "Find all .EXRC and .FORWARD files" -
variable forward -anchor w


        checkbutton .sands_launch.mid.rhoster_f -text "Run SANDS in VERBOSE mode" -variab
le rhoster_f -anchor w
        checkbutton .sands_launch.mid.ngroup -text "Examine CONTENTS of /etc/netgroup" -v
ariable ngroup -anchor w
        checkbutton .sands_launch.mid.majdomo -text "Check MAJORDOMO version" -variable m
ajdomo -anchor w
        checkbutton .sands_launch.mid.complete -text "COMPLETE DIAGNOSTIC" -variable comp
lete -anchor w

        pack .sands_launch.mid.port -in .sands_launch.mid.left -side top -padx 1m -pady 0
m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.serv -in .sands_launch.mid.left -side top -padx 1m -pady 0
m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.equiv -in .sands_launch.mid.left -side top -padx 1m -pady
0m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.rhoster -in .sands_launch.mid.left -side top -padx 1m -pad
y 0m -ipadx 2m -ipady 1m


        pack .sands_launch.mid.grower -in .sands_launch.mid.mid -side top -padx 1m -pady
0m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.file_perms -in .sands_launch.mid.mid -side top -padx 1m -p
ady 0m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.xwind -in .sands_launch.mid.mid -side top -padx 1m -pady 0
m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.forward -in .sands_launch.mid.mid -side top -padx 1m -pady
```

```
 0m -ipadx 2m -ipady 1m


        pack .sands_launch.mid.rhoster_f -in .sands_launch.mid.right -side top -padx 1m
-pady 0m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.ngroup -in .sands_launch.mid.right -side top -padx 1m -pa
dy 0m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.majdomo -in .sands_launch.mid.right -side top -padx 1m -p
ady 0m -ipadx 2m -ipady 1m
        pack .sands_launch.mid.complete -in .sands_launch.mid.right -side top -padx 1m -
pady 0m -ipadx 2m -ipady 1m

        message .sands_launch.top.msg -width 5i -text $text -font -Adobe-Times-Medium-R-
Normal-*-18-*
        pack .sands_launch.top.msg -side right -expand 1 -fill both -padx 3m -pady 3m

        if {$bitmap !="" } {
            label .sands_launch.top.bitmap -bitmap $bitmap
            pack .sands_launch.top.bitmap -side left -padx 3m -pady 3m
         }
        set i 0
        foreach but $args {
           button .sands_launch.bot.button$i -text $but -command "set button $i"
           if {$i == $default} {
              frame .sands_launch.bot.default -relief sunken -bd 1
              raise .sands_launch.bot.button$i
              pack .sands_launch.bot.default -side left -expand 1 -padx 3m -pady 3m
              pack .sands_launch.bot.button$i -in .sands_launch.bot.default -side left -
padx 2m -pady 2m -ipadx 2m -ipady 1m
              } else {
              pack .sands_launch.bot.button$i -side left -padx 2m -pady 2m -ipadx 2m -ip
ady 1m
              }
             incr i
           }



         if {$default >= 0} {
              bind .sands_launch <Return> ".sands_launch.bot.button$default flash; set b
utton $default"
           }


        set oldFocus [focus]

        while {1 > 0} {
           set the_command { }
           tkwait variable button

           if {$port == 1} {set the_command [concat $the_command -P]}
           if {$serv == 1} {set the_command [concat $the_command -S]}
           if {$equiv == 1} {set the_command [concat $the_command -E]}
           if {$rhoster_f == 1} {set the_command [concat $the_command -V]}

           if {$grower == 1} {set the_command [concat $the_command -G]}
           if {$xwind == 1} {set the_command [concat $the_command -X]}
           if {$forward == 1} {set the_command [concat $the_command -F]}
           if {$file_perms == 1} {set the_command [concat $the_command -B]}

           if {$rhoster == 1} {set the_command [concat $the_command -R]}
           if {$ngroup == 1} {set the_command [concat $the_command -N]}
           if {$majdomo == 1} {set the_command [concat $the_command -M]}
           if {$complete == 1} {set the_command [concat $the_command -C]}
```

```
           if {$button == 0} {
              cd /SANDS
              destroy .sands_launch
              update idletasks
              exec ./sands $the_command &
              cd /SALSA
              }

           if {$button == 1} {
              sandsview
              destroy .sands_launch
           }

           if {$button == 2} {
              sandsclear
              destroy .sands_launch
           }

           if {$button == 3} {
              destroy .sands_launch
           }

           if {$button == 4} {
              cd /SALSA
              HelpLoad HELP/sands_help "SANDS"
           }

        }
}
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc whoview {} {
       global host people

        toplevel .whoview -class Dialog
        wm title .whoview {Current Logins}
        wm iconname .whoview {LOGINS}

        frame .whoview.bottom -relief raised
        pack .whoview.bottom -side bottom

        text .whoview.results -relief sunken -width 70 -bd 2 -bg white -yscrollcommand ".
whoview.scroll set"

        pack .whoview.results -side left -fill both -pady 2m

        scrollbar .whoview.scroll -command ".whoview.results yview"

        pack .whoview.scroll -side left -fill y -pady 2m

        set people [exec who]
        set host [exec hostname]

        .whoview.results insert end $people
```

```
	label .whoview.label -text "Logins on:"
	entry .whoview.name -width 20 -relief sunken -bd 2 -fg blue -textvariable host

	pack  .whoview.label -in .whoview.bottom -side left
	pack  .whoview.name -in .whoview.bottom -side left

	button .whoview.exit -text {QUIT} -command {destroy .whoview}

	button .whoview.machine -text {Check Current Logins} -command {
	     .whoview.results delete 1.0 end
	     .whoview.results insert end "\n\n\n\n               WORKING....."
	     update idletasks
	     set people [exec su $the_user -c "rsh $host who"]
	     .whoview.results delete 1.0 end
	     .whoview.results insert end $people
	     }


	pack .whoview.exit -in .whoview.bottom -side right
	pack .whoview.machine -in .whoview.bottom -side right -padx 6m

	bind .whoview.name <Return> {
	   .whoview.results delete 1.0 end
	   .whoview.results insert end "\n\n\n\n               WORKING....."
	   update idletasks
	   set people [exec su $the_user -c "rsh $host who"]
	   .whoview.results delete 1.0 end
	   .whoview.results insert end $people
	}

}


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc lastview {} {
global host oldpeople s_month s_day e_month e_day s_monthstr s_daystr e_monthstr e_dayst
r

	toplevel .lastview -class Dialog
	wm title .lastview {Previous Logins}
	wm iconname .lastview {OLD LOGINS}

	frame .lastview.bottom -relief raised -bd 2
	pack .lastview.bottom -side bottom -pady 2m -ipady 1m -ipadx 1m

	frame .lastview.middle -relief ridge -bd 2
	pack .lastview.middle -side bottom -pady 3m -ipady 1m

	text .lastview.results -relief sunken -width 100 -bd 2 -bg white -yscrollcommand
 ".lastview.scroll set"

	pack .lastview.results -side left -fill both -pady 2m

	scrollbar .lastview.scroll -command ".lastview.results yview"

	pack .lastview.scroll -side left -fill y -pady 2m
```

```
	set oldpeople {}
	cd /SALSA

	set host [exec hostname]

	.lastview.results insert end $oldpeople


	frame .lastview.menubar -relief raised -bd 1m
	pack .lastview.menubar -side bottom -in .lastview.middle -fill x -pady 1m -padx 1
m

	menubutton .lastview.menubar.smonth -text "Start Month" -menu .lastview.menubar.s
month.menu
	menubutton .lastview.menubar.sday -text "Start Day" -menu .lastview.menubar.sday.
menu
	menubutton .lastview.menubar.emonth -text "End Month" -menu .lastview.menubar.emo
nth.menu
	menubutton .lastview.menubar.eday -text "End Day" -menu .lastview.menubar.eday.me
nu

	menu .lastview.menubar.smonth.menu

	.lastview.menubar.smonth.menu add command -label {Current} -command {
	    set s_month 99
	    set s_monthstr "Current"
	}

	.lastview.menubar.smonth.menu add command -label {January} -command {
	    set s_month 1
	    set s_monthstr "January"
	}

	.lastview.menubar.smonth.menu add command -label {February} -command {
	    set s_month 2
	    set s_monthstr "February"
	}

	.lastview.menubar.smonth.menu add command -label {March} -command {
	    set s_month 3
	    set s_monthstr "March"
	}

	.lastview.menubar.smonth.menu add command -label {April} -command {
	    set s_month 4
	    set s_monthstr "April"
	}

	.lastview.menubar.smonth.menu add command -label {May} -command {
	    set s_month 5
	    set s_monthstr "May"
	}

	.lastview.menubar.smonth.menu add command -label {June} -command {
	    set s_month 6
	    set s_monthstr "June"
	}

	.lastview.menubar.smonth.menu add command -label {July} -command {
	    set s_month 7
	    set s_monthstr "July"
	}
```

```
.lastview.menubar.smonth.menu add command -label {August} -command {
    set s_month 8
    set s_monthstr "August"
}

.lastview.menubar.smonth.menu add command -label {September} -command {
    set s_month 9
    set s_monthstr "September"
}

.lastview.menubar.smonth.menu add command -label {October} -command {
    set s_month 10
    set s_monthstr "October"
}

.lastview.menubar.smonth.menu add command -label {November} -command {
    set s_month 11
    set s_monthstr "November"
}

.lastview.menubar.smonth.menu add command -label {December} -command {
    set s_month 12
    set s_monthstr "December"
}


menu .lastview.menubar.sday.menu

.lastview.menubar.sday.menu add command -label {Current} -command {
    set s_day 99
    set s_daystr "Current"
}

.lastview.menubar.sday.menu add command -label {1} -command {
    set s_day 1
    set s_daystr "1"
}

.lastview.menubar.sday.menu add command -label {2} -command {
    set s_day 2
    set s_daystr "2"
}

.lastview.menubar.sday.menu add command -label {3} -command {
    set s_day 3
    set s_daystr "3"
}

.lastview.menubar.sday.menu add command -label {4} -command {
    set s_day 4
    set s_daystr "4"
}

.lastview.menubar.sday.menu add command -label {5} -command {
    set s_day 5
    set s_daystr "5"
}

.lastview.menubar.sday.menu add command -label {6} -command {
    set s_day 6
    set s_daystr "6"
}

.lastview.menubar.sday.menu add command -label {7} -command {
    set s_day 7
    set s_daystr "7"
}

.lastview.menubar.sday.menu add command -label {8} -command {
    set s_day 8
    set s_daystr "8"
}

.lastview.menubar.sday.menu add command -label {9} -command {
    set s_day 9
    set s_daystr "9"
}

.lastview.menubar.sday.menu add command -label {10} -command {
    set s_day 10
    set s_daystr "10"
}

.lastview.menubar.sday.menu add command -label {11} -command {
    set s_day 11
    set s_daystr "11"
}

.lastview.menubar.sday.menu add command -label {12} -command {
    set s_day 12
    set s_daystr "12"
}

.lastview.menubar.sday.menu add command -label {13} -command {
    set s_day 13
    set s_daystr "13"
}

.lastview.menubar.sday.menu add command -label {14} -command {
    set s_day 14
    set s_daystr "14"
}

.lastview.menubar.sday.menu add command -label {15} -command {
    set s_day 15
    set s_daystr "15"
}

.lastview.menubar.sday.menu add command -label {16} -command {
    set s_day 16
    set s_daystr "16"
}

.lastview.menubar.sday.menu add command -label {17} -command {
    set s_day 17
    set s_daystr "17"
}

.lastview.menubar.sday.menu add command -label {18} -command {
    set s_day 18
    set s_daystr "18"
}

.lastview.menubar.sday.menu add command -label {19} -command {
    set s_day 19
    set s_daystr "19"
}
```

```
.lastview.menubar.sday.menu add command -label {20} -command {
    set s_day 20
    set s_daystr "20"
}

.lastview.menubar.sday.menu add command -label {21} -command {
    set s_day 21
    set s_daystr "21"
}

.lastview.menubar.sday.menu add command -label {22} -command {
    set s_day 22
    set s_daystr "22"
}

.lastview.menubar.sday.menu add command -label {23} -command {
    set s_day 23
    set s_daystr "23"
}

.lastview.menubar.sday.menu add command -label {24} -command {
    set s_day 24
    set s_daystr "24"
}

.lastview.menubar.sday.menu add command -label {25} -command {
    set s_day 25
    set s_daystr "25"
}

.lastview.menubar.sday.menu add command -label {26} -command {
    set s_day 26
    set s_daystr "26"
}

.lastview.menubar.sday.menu add command -label {27} -command {
    set s_day 27
    set s_daystr "27"
}

.lastview.menubar.sday.menu add command -label {28} -command {
    set s_day 28
    set s_daystr "28"
}

.lastview.menubar.sday.menu add command -label {29} -command {
    set s_day 29
    set s_daystr "29"
}

.lastview.menubar.sday.menu add command -label {30} -command {
    set s_day 30
    set s_daystr "30"
}

.lastview.menubar.sday.menu add command -label {31} -command {
    set s_day 31
    set s_daystr "31"
}

menu .lastview.menubar.eday.menu

.lastview.menubar.eday.menu add command -label {Log Start} -command {
    set e_day 99
    set e_daystr "Log Start"
}

.lastview.menubar.eday.menu add command -label {1} -command {
    set e_day 1
    set e_daystr "1"
}

.lastview.menubar.eday.menu add command -label {2} -command {
    set e_day 2
    set e_daystr "2"
}

.lastview.menubar.eday.menu add command -label {3} -command {
    set e_day 3
    set e_daystr "3"
}

.lastview.menubar.eday.menu add command -label {4} -command {
    set e_day 4
    set e_daystr "4"
}

.lastview.menubar.eday.menu add command -label {5} -command {
    set e_day 5
    set e_daystr "5"
}

.lastview.menubar.eday.menu add command -label {6} -command {
    set e_day 6
    set e_daystr "6"
}

.lastview.menubar.eday.menu add command -label {7} -command {
    set e_day 7
    set e_daystr "7"
}

.lastview.menubar.eday.menu add command -label {8} -command {
    set e_day 8
    set e_daystr "8"
}

.lastview.menubar.eday.menu add command -label {9} -command {
    set e_day 9
    set e_daystr "9"
}

.lastview.menubar.eday.menu add command -label {10} -command {
    set e_day 10
    set e_daystr "10"
}

.lastview.menubar.eday.menu add command -label {11} -command {
    set e_day 11
    set e_daystr "11"
}

.lastview.menubar.eday.menu add command -label {12} -command {
    set e_day 12
    set e_daystr "12"
}

.lastview.menubar.eday.menu add command -label {13} -command {
```

```
    set e_day 13
    set e_daystr "13"
}

.lastview.menubar.eday.menu add command -label {14} -command {
    set e_day 14
    set e_daystr "14"
}

.lastview.menubar.eday.menu add command -label {15} -command {
    set e_day 15
    set e_daystr "15"
}

.lastview.menubar.eday.menu add command -label {16} -command {
    set e_day 16
    set e_daystr "16"
}

.lastview.menubar.eday.menu add command -label {17} -command {
    set e_day 17
    set e_daystr "17"
}

.lastview.menubar.eday.menu add command -label {18} -command {
    set e_day 18
    set e_daystr "18"
}

.lastview.menubar.eday.menu add command -label {19} -command {
    set e_day 19
    set e_daystr "19"
}

.lastview.menubar.eday.menu add command -label {20} -command {
    set e_day 20
    set e_daystr "20"
}

.lastview.menubar.eday.menu add command -label {21} -command {
    set e_day 21
    set e_daystr "21"
}

.lastview.menubar.eday.menu add command -label {22} -command {
    set e_day 22
    set e_daystr "22"
}

.lastview.menubar.eday.menu add command -label {23} -command {
    set e_day 23
    set e_daystr "23"
}

.lastview.menubar.eday.menu add command -label {24} -command {
    set e_day 24
    set e_daystr "24"
}

.lastview.menubar.eday.menu add command -label {25} -command {
    set e_day 25
    set e_daystr "25"
}

.lastview.menubar.eday.menu add command -label {26} -command {
    set e_day 26
    set e_daystr "26"
}

.lastview.menubar.eday.menu add command -label {27} -command {
    set e_day 27
    set e_daystr "27"
}

.lastview.menubar.eday.menu add command -label {28} -command {
    set e_day 28
    set e_daystr "28"
}

.lastview.menubar.eday.menu add command -label {29} -command {
    set e_day 29
    set e_daystr "29"
}

.lastview.menubar.eday.menu add command -label {30} -command {
    set e_day 30
    set e_daystr "30"
}

.lastview.menubar.eday.menu add command -label {31} -command {
    set e_day 31
    set e_daystr "31"
}


menu  .lastview.menubar.emonth.menu

.lastview.menubar.emonth.menu add command -label {Log Start} -command {
    set e_month 99
    set e_monthstr "Log Start"
}

.lastview.menubar.emonth.menu add command -label {January} -command {
    set e_month 1
    set e_monthstr "January"
}

.lastview.menubar.emonth.menu add command -label {February} -command {
    set e_month 2
    set e_monthstr "February"
}

.lastview.menubar.emonth.menu add command -label {March} -command {
    set e_month 3
    set e_monthstr "March"
}

.lastview.menubar.emonth.menu add command -label {April} -command {
    set e_month 4
    set e_monthstr "April"
}

.lastview.menubar.emonth.menu add command -label {May} -command {
    set e_month 5
    set e_monthstr "May"
}

.lastview.menubar.emonth.menu add command -label {June} -command {
```

```
        set e_month 6
        set e_monthstr "June"
    }

    .lastview.menubar.emonth.menu add command -label {July} -command {
        set e_month 7
        set e_monthstr "July"
    }

    .lastview.menubar.emonth.menu add command -label {August} -command {
        set e_month 8
        set e_monthstr "August"
    }

    .lastview.menubar.emonth.menu add command -label {September} -command {
        set e_month 9
        set e_monthstr "September"
    }

    .lastview.menubar.emonth.menu add command -label {October} -command {
        set e_month 10
        set e_monthstr "October"
    }

    .lastview.menubar.emonth.menu add command -label {November} -command {
        set e_month 11
        set e_monthstr "November"
    }

    .lastview.menubar.emonth.menu add command -label {December} -command {
        set e_month 12
        set e_monthstr "December"
    }


    tk_menuBar .lastview.menubar .lastview.menubar.smonth .lastview.menubar.emonth

    focus .lastview.menubar

    pack .lastview.menubar.smonth -side left -padx 15m
    pack .lastview.menubar.sday -side left -padx 15m
    pack .lastview.menubar.emonth -side left -padx 15m
    pack .lastview.menubar.eday -side left -padx 15m


    set e_month 99
    set s_month 99
    set e_day 99
    set s_day 99

    set s_monthstr Current
    set s_daystr Current
    set e_monthstr "Log Start"
    set e_daystr "Log Start"

    label .lastview.smonth_label -width 9 -textvariable s_monthstr -relief sunken -f
g blue
    label .lastview.sday_label -width 9 -textvariable s_daystr -relief sunken -fg bl
ue
    label .lastview.emonth_label -width 9 -textvariable e_monthstr -relief sunken -f
g blue
    label .lastview.eday_label -width 9 -textvariable e_daystr -relief sunken -fg bl
ue
```

```
        label .lastview.smonth -text "Review log FROM month: "
        label .lastview.sday -text "and day: "
        label .lastview.emonth -text "THROUGH month: "
        label .lastview.eday -text "and day: "

        pack .lastview.smonth .lastview.smonth_label .lastview.sday .lastview.sday_label
.lastview.emonth .lastview.emonth_label .lastview.eday .lastview.eday_label -in .lastview
.middle -side left -ipadx 1m -padx 1m -pady 1m


        label .lastview.label -text "Past Logins on:"
        entry .lastview.name -width 20 -relief sunken -bd 2 -fg blue -textvariable host

        pack  .lastview.label -in .lastview.bottom -side left
        pack  .lastview.name -in .lastview.bottom -side left

        button .lastview.exit -text {QUIT} -command {destroy .lastview}

        button .lastview.save -text {SAVE Results} -command {saveit $oldpeople logins.out
}

        button .lastview.machine -text {Check Past Logins} -command {
            .lastview.results delete 1.0 end
            .lastview.results insert end "\n\n\n\n               WORKING....."
            update idletasks
            exec su $the_user -c "rsh $host last > /tmp/doug_last"
            cd /SALSA
            set oldpeople [exec ./myreader -s $s_month -b $s_day -m $e_month -d $e_day]
            .lastview.results delete 1.0 end
            .lastview.results insert end $oldpeople

        }

        pack .lastview.exit -in .lastview.bottom -side right
        pack .lastview.save -in .lastview.bottom -side right
        pack .lastview.machine -in .lastview.bottom -side right -padx 6m

        bind .lastview.name <Return> {
           .lastview.results delete 1.0 end
           .lastview.results insert end "\n\n\n\n               WORKING....."
           update idletasks
           exec su $the_user -c "rsh $host last > /tmp/doug_last"
           cd /SALSA
           set oldpeople [exec ./myreader -s $s_month -b $s_day -m $e_month -d $e_day]
           .lastview.results delete 1.0 end
           .lastview.results insert end $oldpeople
        }

}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc crondelete {} {
global jobnum

      destroy .crontool.bottom
      frame .crontool.newbottom
```

```
      pack .crontool.newbottom -side top -pady 2m -ipady 1m -ipadx 1m

      frame .crontool.newbottom.display1
      pack  .crontool.newbottom.display1 -side top -in .crontool.newbottom -fill both -p
ady 1m -padx 1m

      frame .crontool.newbottom.bottom -bd 1 -relief raised -background black
      pack .crontool.newbottom.bottom -side bottom -fill both

     label .crontool.label -text "Which Job Number?"

     set jobnum 0

     entry .crontool.jobnum -width 4 -relief sunken -bd 2 -fg blue -textvariable jobnum

     pack .crontool.label .crontool.jobnum -in .crontool.newbottom.display1 -padx 2m

    button .crontool.commit -text {DELETE FROM TABLE} -background green -command {
        cd /SALSA
        catch {exec ./crondel $jobnum} error_msg
        destroy .crontool
        crontool
    }

    button .crontool.forget -text {ABORT} -background red -foreground white -command {de
stroy .crontool}

    pack .crontool.commit .crontool.forget -in .crontool.newbottom.bottom -fill both -an
chor s -padx 3m -pady 3m
}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc cronupdate {} {
global job_type job_typestr s_month s_day hour hourstr minute minutestr s_daystr s_month
str ampm ampmstr freq freqstr button port serv equiv grower file_perms xwind forward rho
ster rhoster_f ngroup majdomo complete oldFocus2 the_command target

      destroy .crontool.bottom
      frame .crontool.newbottom
      pack .crontool.newbottom -side top -pady 2m -ipady 1m -ipadx 1m

      frame .crontool.newbottom.display1
      pack  .crontool.newbottom.display1 -side top -in .crontool.newbottom -fill x -pady
 1m -padx 1m

      frame .crontool.newbottom.menubar1 -relief raised -bd 1m
      pack  .crontool.newbottom.menubar1 -side top -in .crontool.newbottom -fill x -pady
 1m -padx 1m

      frame .crontool.newbottom.display2
      pack  .crontool.newbottom.display2 -side top -in .crontool.newbottom -fill x -pady
 1m -padx 1m

      frame .crontool.newbottom.menubar2 -relief raised -bd 1m
      pack  .crontool.newbottom.menubar2 -side top -in .crontool.newbottom -fill x -pady
 1m -padx 1m

      frame .crontool.newbottom.display3
      pack  .crontool.newbottom.display3 -side top -in .crontool.newbottom -fill x -pady
1m -padx 1m

      frame .crontool.newbottom.menubar3 -relief raised -bd 1m
      pack  .crontool.newbottom.menubar3 -side top -in .crontool.newbottom -fill x -pady
1m -padx 1m

           frame .crontool.newbottom.display4
           pack  .crontool.newbottom.display4 -side top -in .crontool.newbottom -fill x
-pady 1m -padx 1m

           frame .crontool.newbottom.menubar4 -relief raised -bd 1m
           pack  .crontool.newbottom.menubar4 -side top -in .crontool.newbottom -fill x
-pady 1m -padx 1m



      frame .crontool.newbottom.bottom -bd 1 -relief raised -background black
      pack .crontool.newbottom.bottom -side bottom -fill both


      set e_month 99
      set job_type 99
      set minute 00
      set hour 00
      set ampm 00
      set s_day 1
      set freq 1

        set ampmstr "AM"
        set job_typestr "None"
        set hourstr "12"
        set minutestr "00"
        set freqstr "Daily"

        set s_daystr "12"
        set e_monthstr "Log Start"


        label .crontool.jobtype_label -width 9 -textvariable job_typestr -relief sunken -
fg blue
        label .crontool.hour_label -width 3 -textvariable hourstr -relief sunken -fg blue
        label .crontool.minute_label -width 3 -textvariable minutestr -relief sunken -fg
blue
        label .crontool.ampm_label -width 3 -textvariable ampmstr -relief sunken -fg blue
        label .crontool.freq_label -width 8 -textvariable freqstr -relief sunken -fg blue

        label .crontool.emonth_label -width 9 -textvariable e_monthstr -relief sunken -fg
 blue
        label .crontool.eday_label -width 9 -textvariable hourstr -relief sunken -fg blue

        label .crontool.jobtype -text "The Job Type: "
        label .crontool.hour -text "Hour: "
        label .crontool.minute -text "Minute: "
        label .crontool.freq -text "How often: "

        label .crontool.smonth -text "THROUGH month: "


        pack .crontool.jobtype .crontool.jobtype_label -in .crontool.newbottom.display1 -
side left
        pack .crontool.hour .crontool.hour_label -in .crontool.newbottom.display2 -side l
```

```
eft
        pack .crontool.minute .crontool.minute_label -in .crontool.newbottom.display2 -s
ide left
        pack .crontool.ampm_label -in .crontool.newbottom.display2 -side left -padx 3m
        pack .crontool.freq .crontool.freq_label -in .crontool.newbottom.display3 -side
left

        menubutton .crontool.newbottom.menubar1.jobtype -text "Job Type" -menu .crontool
.newbottom.menubar1.jobtype.menu

        menubutton .crontool.newbottom.menubar2.hour -text "Hour" -menu .crontool.newbot
tom.menubar2.hour.menu
        menubutton .crontool.newbottom.menubar2.minute -text "Minute" -menu .crontool.ne
wbottom.menubar2.minute.menu
        menubutton .crontool.newbottom.menubar2.ampm -text "AM / PM" -menu .crontool.new
bottom.menubar2.ampm.menu
        menubutton .crontool.newbottom.menubar3.freq -text "Frequency" -menu .crontool.n
ewbottom.menubar3.freq.menu

        menu .crontool.newbottom.menubar1.jobtype.menu

        .crontool.newbottom.menubar1.jobtype.menu add command -label {None} -command {
            set job_type 99
            set job_typestr "None"
        }

        .crontool.newbottom.menubar1.jobtype.menu add command -label {SANDS} -command {

            if {$job_type == 1} {
               destroy .crontool.left
               destroy .crontool.mid
               destroy .crontool.right
               destroy .crontool.port
               destroy .crontool.serv
               destroy .crontool.equiv
               destroy .crontool.rhoster
               destroy .crontool.grower
               destroy .crontool.xwind
               destroy .crontool.file_perms
               destroy .crontool.forward
               destroy .crontool.rhoster_f
               destroy .crontool.ngroup
               destroy .crontool.majdomo
               destroy .crontool.complete

            }

          if {$job_type == 4} {
              destroy .crontool.left
              destroy .crontool.crentry
              destroy .crontool.crlabel
          }

          set job_type 1
          set job_typestr "SANDS"

          frame .crontool.left -relief raised -bd 1
          pack .crontool.left -fill both -side top

          frame .crontool.mid -relief raised -bd 1
          pack .crontool.mid -fill both -side top

          frame .crontool.right -relief raised -bd 1
          pack .crontool.right -fill both -side top

          checkbutton .crontool.port -text "Examine PORTMAPPER Services" -variable port -an
chor w
          checkbutton .crontool.serv -text "Examine SERVICES in inetd.conf" -variable serv
-anchor w
          checkbutton .crontool.equiv -text "Examine Hosts in HOSTS.EQUIV" -variable equiv
-anchor w
          checkbutton .crontool.rhoster -text "Find all .RHOSTS files" -variable rhoster -a
nchor w



          checkbutton .crontool.grower -text "Do NOT avoid NFS mounted file systems" -varia
ble grower -anchor w
          checkbutton .crontool.xwind -text "Check XWIN security" -variable xwind -anchor w

          checkbutton .crontool.file_perms -text "Check FILE PERMISSIONS" -variable file_pe
rms -anchor w
          checkbutton .crontool.forward -text "Find all .EXRC and .FORWARD files" -variable
 forward -anchor w


          checkbutton .crontool.rhoster_f -text "Run SANDS in VERBOSE mode" -variable rhost
er_f -anchor w
          checkbutton .crontool.ngroup -text "Examine CONTENTS of /etc/netgroup" -variable
ngroup -anchor w
          checkbutton .crontool.majdomo -text "Check MAJORDOMO version" -variable majdomo -
anchor w
          checkbutton .crontool.complete -text "COMPLETE DIAGNOSTIC" -variable complete -an
chor w

          pack .crontool.port -in .crontool.left -side top -padx 1m -pady 0m -ipadx 2m -ipa
dy 1m
          pack .crontool.serv -in .crontool.left -side top -padx 1m -pady 0m -ipadx 2m -ipa
dy 1m
          pack .crontool.equiv -in .crontool.left -side top -padx 1m -pady 0m -ipadx 2m -ip
ady 1m
          pack .crontool.rhoster -in .crontool.left -side top -padx 1m -pady 0m -ipadx 2m -
ipady 1m


          pack .crontool.grower -in .crontool.mid -side top -padx 1m -pady 0m -ipadx 2m -ip
ady 1m
          pack .crontool.file_perms -in .crontool.mid -side top -padx 1m -pady 0m -ipadx 2m
 -ipady 1m
          pack .crontool.xwind -in .crontool.mid -side top -padx 1m -pady 0m -ipadx 2m -ipa
dy 1m
          pack .crontool.forward -in .crontool.mid -side top -padx 1m -pady 0m -ipadx 2m -i
pady 1m


          pack .crontool.rhoster_f -in .crontool.right -side top -padx 1m -pady 0m -ipadx 2
m -ipady 1m
          pack .crontool.ngroup -in .crontool.right -side top -padx 1m -pady 0m -ipadx 2m -
ipady 1m
          pack .crontool.majdomo -in .crontool.right -side top -padx 1m -pady 0m -ipadx 2m
-ipady 1m
          pack .crontool.complete -in .crontool.right -side top -padx 1m -pady 0m -ipadx 2m
 -ipady 1m

          }

          .crontool.newbottom.menubar1.jobtype.menu add command -label {TIGER} -command {
              if {$job_type == 1} {
```

```
        destroy .crontool.left
        destroy .crontool.mid
        destroy .crontool.right

        destroy .crontool.port
        destroy .crontool.serv
        destroy .crontool.equiv
        destroy .crontool.rhoster
        destroy .crontool.grower
        destroy .crontool.xwind
        destroy .crontool.file_perms
        destroy .crontool.forward
        destroy .crontool.rhoster_f
        destroy .crontool.ngroup
        destroy .crontool.majdomo
        destroy .crontool.complete

    }

  if {$job_type == 4} {
      destroy .crontool.left
      destroy .crontool.crentry
      destroy .crontool.crlabel
  }

    set job_type 2
    set job_typestr "TIGER"
}

.crontool.newbottom.menubar1.jobtype.menu add command -label {COPS} -command {

    if {$job_type == 1} {
        destroy .crontool.left
        destroy .crontool.mid
        destroy .crontool.right
        destroy .crontool.port
        destroy .crontool.serv
        destroy .crontool.equiv
        destroy .crontool.rhoster
        destroy .crontool.grower
        destroy .crontool.xwind
        destroy .crontool.file_perms
        destroy .crontool.forward
        destroy .crontool.rhoster_f
        destroy .crontool.ngroup
        destroy .crontool.majdomo
        destroy .crontool.complete

    }

  if {$job_type == 4} {
      destroy .crontool.left
      destroy .crontool.crentry
      destroy .crontool.crlabel
  }

    set job_type 3
    set job_typestr "COPS"
}

.crontool.newbottom.menubar1.jobtype.menu add command -label {CRACK} -command {

    if {$job_type == 1} {
        destroy .crontool.left
        destroy .crontool.mid
        destroy .crontool.right
        destroy .crontool.port
        destroy .crontool.serv
        destroy .crontool.equiv
        destroy .crontool.rhoster
        destroy .crontool.grower
        destroy .crontool.xwind
        destroy .crontool.file_perms
        destroy .crontool.forward
        destroy .crontool.rhoster_f
        destroy .crontool.ngroup
        destroy .crontool.majdomo
        destroy .crontool.complete

    }

  if {$job_type == 4} {
      destroy .crontool.left
      destroy .crontool.crentry
      destroy .crontool.crlabel
  }

    set job_type 4
    set job_typestr "CRACK"

  frame .crontool.left -relief raised -bd 1
  pack .crontool.left -fill both -side top


    label .crontool.crlabel -text "Input the password file name: "

    entry .crontool.crentry -textvariable target -fg blue

    pack .crontool.crlabel .crontool.crentry -in .crontool.left -side left -padx 1
m -pady 2m

}



tk_menuBar .crontool.newbottom.menubar1 .crontool.newbottom.menubar1.jobtype

focus .crontool.newbottom.menubar1



menu .crontool.newbottom.menubar2.ampm.menu

.crontool.newbottom.menubar2.ampm.menu add command -label {AM} -command {
    set ampm 0
    set ampmstr "AM"
}

.crontool.newbottom.menubar2.ampm.menu add command -label {PM} -command {
    set ampm 12
    set ampmstr "PM"
}



menu .crontool.newbottom.menubar2.hour.menu

.crontool.newbottom.menubar2.hour.menu add command -label {1} -command {
```

```
    set hour 1
    set hourstr "1"
}

.crontool.newbottom.menubar2.hour.menu add command -label {2} -command {
    set hour 2
    set hourstr "2"
}

.crontool.newbottom.menubar2.hour.menu add command -label {3} -command {
    set hour 3
    set hourstr "3"
}

.crontool.newbottom.menubar2.hour.menu add command -label {4} -command {
    set hour 4
    set hourstr "4"
}

.crontool.newbottom.menubar2.hour.menu add command -label {5} -command {
    set hour 5
    set hourstr 5
}


.crontool.newbottom.menubar2.hour.menu add command -label {6} -command {
    set hour 6
    set hourstr "6"
}

.crontool.newbottom.menubar2.hour.menu add command -label {7} -command {
    set hour 7
    set hourstr "7"
}

.crontool.newbottom.menubar2.hour.menu add command -label {8} -command {
    set hour 8
    set hourstr "8"
}

.crontool.newbottom.menubar2.hour.menu add command -label {9} -command {
    set hour 9
    set hourstr "9"
}

.crontool.newbottom.menubar2.hour.menu add command -label {10} -command {
    set hour 10
    set hourstr "10"
}

.crontool.newbottom.menubar2.hour.menu add command -label {11} -command {
    set hour 11
    set hourstr "11"
}

.crontool.newbottom.menubar2.hour.menu add command -label {12} -command {
    set hour 12
    set hourstr "12"
}



menu .crontool.newbottom.menubar2.minute.menu
```

```
.crontool.newbottom.menubar2.minute.menu add command -label {00} -command {
    set minute 00
    set minutestr "00"
}

.crontool.newbottom.menubar2.minute.menu add command -label {10} -command {
    set minute 10
    set minutestr "10"
}

.crontool.newbottom.menubar2.minute.menu add command -label {20} -command {
    set minute 20
    set minutestr "20"
}

.crontool.newbottom.menubar2.minute.menu add command -label {30} -command {
    set minute 30
    set minutestr "30"
}

.crontool.newbottom.menubar2.minute.menu add command -label {40} -command {
    set minute 40
    set minutestr "40"
}

.crontool.newbottom.menubar2.minute.menu add command -label {50} -command {
    set minute 50
    set minutestr "50"
}



tk_menuBar .crontool.newbottom.menubar2 .crontool.newbottom.menubar2.hour .cronto
ol.newbottom.menubar2.minute .crontool.newbottom.menubar2.ampm

focus .crontool.newbottom.menubar2


menu .crontool.newbottom.menubar3.freq.menu

.crontool.newbottom.menubar3.freq.menu add command -label {Daily} -command {

   if {$freq == 1} {
       destroy .crontool.newbottom.display4
       destroy .crontool.newbottom.menubar4
   }

   if {$freq == 2} {
       destroy .crontool.newbottom.display4
       destroy .crontool.newbottom.menubar4
       destroy .crontool.sday
       destroy .crontool.sday_label
   }

   if {$freq == 3} {
       destroy .crontool.newbottom.display4
       destroy .crontool.newbottom.menubar4
       destroy .crontool.mday
       destroy .crontool.mday_label
   }

    frame .crontool.newbottom.display4
    pack  .crontool.newbottom.display4 -side top -in .crontool.newbottom -fill x
```

```
-pady 1m -padx 1m

            frame .crontool.newbottom.menubar4 -relief raised -bd 1m
            pack  .crontool.newbottom.menubar4 -side top -in .crontool.newbottom -fill x
 -pady 1m -padx 1m

           set freq 1
           set freqstr "Daily"

        }

        .crontool.newbottom.menubar3.freq.menu add command -label {Weekly} -command {

           if {$freq == 1} {
               destroy .crontool.newbottom.display4
               destroy .crontool.newbottom.menubar4
           }

           if {$freq == 2} {
               destroy .crontool.newbottom.display4
               destroy .crontool.newbottom.menubar4
               destroy .crontool.sday
               destroy .crontool.sday_label
           }

           if {$freq == 3} {
               destroy .crontool.newbottom.display4
               destroy .crontool.newbottom.menubar4
               destroy .crontool.mday
               destroy .crontool.mday_label
           }


           set freq 2
           set freqstr "Weekly"

            frame .crontool.newbottom.display4
            pack  .crontool.newbottom.display4 -side top -in .crontool.newbottom -fill x
 -pady 1m -padx 1m

            frame .crontool.newbottom.menubar4 -relief raised -bd 1m
            pack  .crontool.newbottom.menubar4 -side top -in .crontool.newbottom -fill x
 -pady 1m -padx 1m

            set s_daystr "Monday"
            set s_day 1

            label .crontool.sday -text "Day: "
            label .crontool.sday_label -width 10 -textvariable s_daystr -relief sunken -
fg blue
            pack .crontool.sday .crontool.sday_label -in .crontool.newbottom.display4 -s
ide left

        menubutton .crontool.newbottom.menubar4.day -text "Which Day" -menu .crontool.ne
wbottom.menubar4.day.menu

        menu .crontool.newbottom.menubar4.day.menu

        .crontool.newbottom.menubar4.day.menu add command -label {Monday} -command {
            set s_day 1
            set s_daystr "Monday"
        }

        .crontool.newbottom.menubar4.day.menu add command -label {Tuesday} -command {
            set s_day 2
            set s_daystr "Tuesday"
        }

        .crontool.newbottom.menubar4.day.menu add command -label {Wednesday} -command {
            set s_day 3
            set s_daystr "Wednesday"
        }

        .crontool.newbottom.menubar4.day.menu add command -label {Thursday} -command {
            set s_day 4
            set s_daystr "Thursday"
        }

        .crontool.newbottom.menubar4.day.menu add command -label {Friday} -command {
            set s_day 5
            set s_daystr "Friday"
        }

        .crontool.newbottom.menubar4.day.menu add command -label {Saturday} -command {
            set s_day 6
            set s_daystr "Saturday"
        }

        .crontool.newbottom.menubar4.day.menu add command -label {Sunday} -command {
            set s_day 0
            set s_daystr "Sunday"
        }


        tk_menuBar .crontool.newbottom.menubar4 .crontool.newbottom.menubar4.day

        focus .crontool.newbottom.menubar4

        pack .crontool.newbottom.menubar4.day -side left -padx 2m


        }

        .crontool.newbottom.menubar3.freq.menu add command -label {Montly} -command {
           if {$freq == 1} {
               destroy .crontool.newbottom.display4
               destroy .crontool.newbottom.menubar4
           }

           if {$freq == 2} {
               destroy .crontool.newbottom.display4
               destroy .crontool.newbottom.menubar4
               destroy .crontool.sday
               destroy .crontool.sday_label
           }

           if {$freq == 3} {
               destroy .crontool.newbottom.display4
               destroy .crontool.newbottom.menubar4
               destroy .crontool.mday
               destroy .crontool.mday_label
           }


            set freq 3
            set freqstr "Monthly"

            frame .crontool.newbottom.display4
```

```
            pack  .crontool.newbottom.display4 -side top -in .crontool.newbottom -fill x
 -pady 1m -padx 1m

            frame .crontool.newbottom.menubar4 -relief raised -bd 1m
            pack  .crontool.newbottom.menubar4 -side top -in .crontool.newbottom -fill x
 -pady 1m -padx 1m

            set s_monthstr "1"
            set s_month 1

            label .crontool.mday -text "Day of Month: "
            label .crontool.mday_label -width 10 -textvariable s_monthstr -relief sunken
 -fg blue
            pack .crontool.mday .crontool.mday_label -in .crontool.newbottom.display4 -s
ide left

        menubutton .crontool.newbottom.menubar4.month -text "The Date" -menu .crontool.n
ewbottom.menubar4.month.menu

        menu .crontool.newbottom.menubar4.month.menu



        .crontool.newbottom.menubar4.month.menu add command -label {1} -command {
            set s_month 1
            set s_monthstr "1"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {2} -command {
            set s_month 2
            set s_monthstr "2"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {3} -command {
            set s_month 3
            set s_monthstr "3"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {4} -command {
            set s_month 4
            set s_monthstr "4"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {5} -command {
            set s_month 5
            set s_monthstr "5"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {6} -command {
            set s_month 6
            set s_monthstr "6"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {7} -command {
            set s_month 7
            set s_monthstr "7"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {8} -command {
            set s_month 8
            set s_monthstr "8"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {9} -command {
            set s_month 9
            set s_monthstr "9"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {10} -command {
            set s_month 10
            set s_monthstr "10"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {11} -command {
            set s_month 11
            set s_monthstr "11"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {12} -command {
            set s_month 12
            set s_monthstr "12"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {13} -command {
            set s_month 13
            set s_monthstr "13"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {14} -command {
            set s_month 14
            set s_monthstr "14"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {15} -command {
            set s_month 15
            set s_monthstr "15"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {16} -command {
            set s_month 16
            set s_monthstr "16"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {17} -command {
            set s_month 17
            set s_monthstr "17"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {18} -command {
            set s_month 18
            set s_monthstr "18"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {19} -command {
            set s_month 19
            set s_monthstr "19"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {20} -command {
            set s_month 20
            set s_monthstr "20"
        }

        .crontool.newbottom.menubar4.month.menu add command -label {21} -command {
            set s_month 21
            set s_monthstr "21"
        }
```

```
.crontool.newbottom.menubar4.month.menu add command -label {22} -command {
    set s_month 22
    set s_monthstr "22"
}

.crontool.newbottom.menubar4.month.menu add command -label {23} -command {
    set s_month 23
    set s_monthstr "23"
}

.crontool.newbottom.menubar4.month.menu add command -label {24} -command {
    set s_month 24
    set s_monthstr "24"
}

.crontool.newbottom.menubar4.month.menu add command -label {25} -command {
    set s_month 25
    set s_monthstr "25"
}

.crontool.newbottom.menubar4.month.menu add command -label {26} -command {
    set s_month 26
    set s_monthstr "26"
}

.crontool.newbottom.menubar4.month.menu add command -label {27} -command {
    set s_month 27
    set s_monthstr "27"
}

.crontool.newbottom.menubar4.month.menu add command -label {28} -command {
    set s_month 28
    set s_monthstr "28"
}

.crontool.newbottom.menubar4.month.menu add command -label {29} -command {
    set s_month 29
    set s_monthstr "29"
}

.crontool.newbottom.menubar4.month.menu add command -label {30} -command {
    set s_month 30
    set s_monthstr "30"
}

.crontool.newbottom.menubar4.month.menu add command -label {31} -command {
    set s_month 31
    set s_monthstr "31"
}


tk_menuBar .crontool.newbottom.menubar4 .crontool.newbottom.menubar4.month

focus .crontool.newbottom.menubar4

pack .crontool.newbottom.menubar4.month -side left -padx 2m

}

tk_menuBar .crontool.newbottom.menubar3 .crontool.newbottom.menubar3.freq

focus .crontool.newbottom.menubar3
```

```
        pack .crontool.newbottom.menubar1.jobtype -side left
        pack .crontool.newbottom.menubar2.hour -side left -padx 2m
        pack .crontool.newbottom.menubar2.minute -side left -padx 2m
        pack .crontool.newbottom.menubar2.ampm -side left -padx 2m
        pack .crontool.newbottom.menubar3.freq -side left -padx 2m


        # pack .crontool.newbottom.menubar1.emonth -side left -padx 15m
        # pack .crontool.newbottom.menubar1.eday -side left -padx 15m


        button .crontool.commit -text {COMMIT TO TABLE} -background green -command {
            if {($ampm == 0) && ($hour == 12)} {
                set hour 00
            }

             if {$freq == 1} {
                 if {$job_type == 1} {
                 set the_command { }

                if {$port == 1} {set the_command [concat $the_command -P]}
                if {$serv == 1} {set the_command [concat $the_command -S]}
                if {$equiv == 1} {set the_command [concat $the_command -E]}
                if {$rhoster_f == 1} {set the_command [concat $the_command -V]}

                if {$grower == 1} {set the_command [concat $the_command -G]}
                if {$xwind == 1} {set the_command [concat $the_command -X]}
                if {$forward == 1} {set the_command [concat $the_command -F]}
                if {$file_perms == 1} {set the_command [concat $the_command -B]}

                if {$rhoster == 1} {set the_command [concat $the_command -R]}
                if {$ngroup == 1} {set the_command [concat $the_command -N]}
                if {$majdomo == 1} {set the_command [concat $the_command -M]}
                if {$complete == 1} {set the_command [concat $the_command -C]}

                    exec cp /var/spool/cron/crontabs/root ./temproot
                    exec echo "$minute [expr $hour + $ampm] * * * cd /SANDS; ./sands $th
e_command" >> ./temproot
                    exec crontab ./temproot
                    exec rm temproot
                }
                if {$job_type == 2} {
                    exec cp /var/spool/cron/crontabs/root ./temproot
                    exec echo "$minute [expr $hour + $ampm] * * * cd /TIGER/tiger-2.2.3;
 ./tiger" >> ./temproot
                    exec crontab ./temproot
                    exec rm temproot

                }
                if {$job_type == 3} {
                    exec cp /var/spool/cron/crontabs/root ./temproot
                    exec echo "$minute [expr $hour + $ampm] * * * cd /COPS/cops_104; ./c
ops -v -s . -b cops.err" >> ./temproot
                    exec crontab ./temproot
                    exec rm temproot
                }
                if {$job_type == 4} {
                    exec cp /var/spool/cron/crontabs/root ./temproot
                    exec echo "$minute [expr $hour + $ampm] * * * cd /CRACK; ./Crack $ta
rget" >> ./temproot
                    exec crontab ./temproot
                    exec rm temproot
                }
```

```
            destroy .crontool
            crontool
        }


        if {$freq == 2} {
            if {$job_type == 1} {
            set the_command { }

            if {$port == 1} {set the_command [concat $the_command -P]}
            if {$serv == 1} {set the_command [concat $the_command -S]}
            if {$equiv == 1} {set the_command [concat $the_command -E]}
            if {$rhoster_f == 1} {set the_command [concat $the_command -V]}

            if {$grower == 1} {set the_command [concat $the_command -G]}
            if {$xwind == 1} {set the_command [concat $the_command -X]}
            if {$forward == 1} {set the_command [concat $the_command -F]}
            if {$file_perms == 1} {set the_command [concat $the_command -B]}

            if {$rhoster == 1} {set the_command [concat $the_command -R]}
            if {$ngroup == 1} {set the_command [concat $the_command -N]}
            if {$majdomo == 1} {set the_command [concat $the_command -M]}
            if {$complete == 1} {set the_command [concat $the_command -C]}

                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] * * $s_day cd /SANDS; ./san
ds $the_command" >> ./temproot
                exec crontab ./temproot
                exec rm temproot

            }
            if {$job_type == 2} {
                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] * * $s_day cd /TIGER/tiger-
2.2.3; ./tiger" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }
            if {$job_type == 3} {
                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] * * $s_day cd /COPS/cops_10
4; ./cops -v -s . -b cops.err" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }
            if {$job_type == 4} {
                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] * * $s_day cd /CRACK; ./Cra
ck $target" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }

            destroy .crontool
            crontool

        }


        if {$freq == 3} {
            if {$job_type == 1} {
            set the_command { }

            if {$port == 1} {set the_command [concat $the_command -P]}
            if {$serv == 1} {set the_command [concat $the_command -S]}
            if {$equiv == 1} {set the_command [concat $the_command -E]}
            if {$rhoster_f == 1} {set the_command [concat $the_command -V]}

            if {$grower == 1} {set the_command [concat $the_command -G]}
            if {$xwind == 1} {set the_command [concat $the_command -X]}
            if {$forward == 1} {set the_command [concat $the_command -F]}
            if {$file_perms == 1} {set the_command [concat $the_command -B]}

            if {$rhoster == 1} {set the_command [concat $the_command -R]}
            if {$ngroup == 1} {set the_command [concat $the_command -N]}
            if {$majdomo == 1} {set the_command [concat $the_command -M]}
            if {$complete == 1} {set the_command [concat $the_command -C]}

                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] $s_month * * cd /SANDS; ./sa
nds $the_command" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }
            if {$job_type == 2} {
                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] $s_month * * cd /TIGER/tiger
-2.2.3; ./tiger" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }

            if {$job_type == 3} {
                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] $s_month * * cd /COPS/cops_1
04; ./cops -v -s . -b cops.err" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }

            if {$job_type == 4} {
                exec cp /var/spool/cron/crontabs/root ./temproot
                exec echo "$minute [expr $hour + $ampm] $s_month * * cd /CRACK; ./Cr
ack $target" >> ./temproot
                exec crontab ./temproot
                exec rm temproot
            }

            destroy .crontool
            crontool

        }
    }


    button .crontool.forget -text {ABORT} -background red -foreground white -command
{destroy .crontool}

    pack .crontool.commit .crontool.forget -in .crontool.newbottom.bottom -fill both
-anchor s -padx 3m -pady 3m

# .crontool.sday .crontool.sday_label .crontool.emonth .crontool.emonth_label .crontool.e
day .crontool.eday_label -in .crontool.middle -side left -ipadx 1m -padx 1m -pady 1m

}
```

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx


proc crontool {} {
global host table_contents

        toplevel .crontool -class Dialog
        set host [exec hostname]
        wm title .crontool "UPDATE CRON TABLE for $host"
        wm iconname .crontool {CRON TOOL}

        frame .crontool.bottom
        pack .crontool.bottom -side bottom -pady 2m -ipady 1m -ipadx 1m

        frame .crontool.middle -relief ridge -bd 2
#       pack .crontool.middle -side bottom -pady 3m -ipady 1m

        text .crontool.results -relief sunken -width 70 -bd 2 -bg white -yscrollcommand
".crontool.scroll set"

        pack .crontool.results -side left -fill both -pady 2m

        scrollbar .crontool.scroll -command ".crontool.results yview"

        pack .crontool.scroll -side left -fill y -pady 2m

        cd /SALSA
        set table_contents [exec ./croner]

        .crontool.results insert end $table_contents

        button .crontool.exit -text {QUIT} -command {destroy .crontool}
        button .crontool.help -text {HELP} -command {
                cd /SALSA
                HelpLoad HELP/planner_help "The PERIOD PLANNER"
        }

        button .crontool.add -text {ADD JOB} -command {cronupdate}

        button .crontool.remove -text {REMOVE JOB} -command {crondelete}


        pack .crontool.help -in .crontool.bottom -side right -padx 2m
        pack .crontool.exit -in .crontool.bottom -side right -padx 2m
        pack .crontool.remove -in .crontool.bottom -side right -padx 2m
        pack .crontool.add -in .crontool.bottom -side right -padx 2m


}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
```

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc findchanges {} {
       global host people choice the_dir starter ender p_st p_end

        toplevel .findchanges -class Dialog
        wm title .findchanges {FIND CHANGED / ACCESSED Files}
        wm iconname .findchanges {CHANGES}

        frame .findchanges.btm -relief raised -bd 3
        pack .findchanges.btm -side bottom -fill both

        frame .findchanges.bottom -relief ridge -bd 3
        pack .findchanges.bottom -side bottom -fill both

        frame .findchanges.bottom.left -relief ridge -bd 1
        pack .findchanges.bottom.left  -side left -fill both -in .findchanges.bottom

        frame .findchanges.bottom.left.left -relief ridge
        pack .findchanges.bottom.left.left  -side left -fill both -in .findchanges.bottom
.left

        frame .findchanges.bottom.left.right -relief ridge
        pack .findchanges.bottom.left.right  -side left -fill both -in .findchanges.botto
m.left

        frame .findchanges.bottom.right -relief ridge -bd 1
        pack .findchanges.bottom.right  -side right -fill both -in .findchanges.bottom

        text .findchanges.results -relief sunken -width 90 -bd 2 -bg white -yscrollcomman
d ".findchanges.scroll set"

        pack .findchanges.results -side left -fill both -pady 2m

        scrollbar .findchanges.scroll -command ".findchanges.results yview"

        pack .findchanges.scroll -side left -fill y -pady 2m

        set people {}
        set host [exec hostname]

        set choice 3
        set the_dir "/"
        set starter 0
        set ender 0

        set p_st  0
        set p_end 0

        .findchanges.results insert end $people

        label .findchanges.label -text "Files on :"
        label .findchanges.name -width 20 -relief sunken -bd 2 -fg blue -textvariable hos
t

        label .findchanges.label2 -text "Directory:"
        entry .findchanges.name2 -width 20 -relief sunken -bd 2 -fg blue -textvariable th
e_dir
```

```
        label .findchanges.label3 -text "The Time Period (Number of Days)"

        label .findchanges.label4 -text "From:"
        entry .findchanges.name4 -width 5 -relief sunken -bd 2 -fg blue -textvariable st
arter

        label .findchanges.label5 -text "To:"
        entry .findchanges.name5 -width 5 -relief sunken -bd 2 -fg blue -textvariable en
der


        pack  .findchanges.label -in .findchanges.bottom.left.left -pady 2m -padx 1m
        pack  .findchanges.name -in .findchanges.bottom.left.right -pady 2m -padx 1m

        pack  .findchanges.label2 -in .findchanges.bottom.left.left -pady 2m -padx 1m
        pack  .findchanges.name2 -in .findchanges.bottom.left.right -side right -pady 2m
 -padx 1m

        button .findchanges.exit -text {QUIT} -command {destroy .findchanges}
        button .findchanges.save -text {SAVE REPORT} -command {saveit $people changes.ou
t}

        button .findchanges.machine -relief groove -text {FIND FILES} -command {
             .findchanges.results delete 1.0 end
             .findchanges.results insert end "\n\n\n\n               WORKING....."
             update idletasks

             if {$choice == 1} {
                 set p_st [expr $starter - 1]
                 set p_end [expr $ender + 1]
                 set people [exec find $the_dir -ctime +$p_st -ctime -$p_end]
             }

             if {$choice == 2} {
                 set p_st [expr $starter - 1]
                 set p_end [expr $ender + 1]
                 set people [exec find $the_dir -atime +$p_st -atime -$p_end]

             }

             if {$choice == 3} {
                 set p_st [expr $starter - 1]
                 set p_end [expr $ender + 1]
                 set people [exec find $the_dir -ctime +$p_st -ctime -$p_end]
                 set people [concat $people [exec find $the_dir -atime +$p_st -atime -$p
_end]]
             }

             .findchanges.results delete 1.0 end
             .findchanges.results insert end $people
         }


        radiobutton .findchanges.c -text "Created / Modified Files" -variable choice -an
chor e -value 1
        radiobutton .findchanges.a -text "Accessed Files           " -variable choice -a
nchor e -value 2
        radiobutton .findchanges.both -text "Accessed and Changed   " -variable choice -
anchor e -value 3

        pack .findchanges.label3 -in .findchanges.bottom -side top
        pack .findchanges.label4 .findchanges.name4 .findchanges.label5 .findchanges.nam
e5 -in .findchanges.bottom -side left

        pack .findchanges.machine -in .findchanges.bottom -side bottom -fill x -padx 6m -
pady 2m


        pack .findchanges.c .findchanges.a .findchanges.both -in .findchanges.bottom.righ
t -side top

        pack .findchanges.save .findchanges.exit -in .findchanges.btm -side left  -pady 2
m -padx 2m

        bind .findchanges.name <Return> {
             focus .findchanges.name2
             }
        bind .findchanges.name2 <Return> {
             focus .findchanges.name4
             }
        bind .findchanges.name4 <Return> {
             focus .findchanges.name5
             }

        bind .findchanges.name5 <Return> {
             .findchanges.results delete 1.0 end
             .findchanges.results insert end "\n\n\n\n               WORKING....."
             update idletasks

             if {$choice == 1} {
                 set p_st [expr $starter - 1]
                 set p_end [expr $ender + 1]
                 set people [exec find $the_dir -ctime +$p_st -ctime -$p_end]
             }

             if {$choice == 2} {
                 set p_st [expr $starter - 1]
                 set p_end [expr $ender + 1]
                 set people [exec find $the_dir -atime +$p_st -atime -$p_end]

             }

             if {$choice == 3} {
                 set p_st [expr $starter - 1]
                 set p_end [expr $ender + 1]
                 set people [exec find $the_dir -ctime +$p_st -ctime -$p_end]
                 set people [concat $people [exec find $the_dir -atime +$p_st -atime -$p_
end]]
             }

             .findchanges.results delete 1.0 end
             .findchanges.results insert end $people

         }

}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
```

```
proc saveit {mystuff to_file} {
global which filename oldFocus4
          toplevel .saver -class Dialog
          wm title .saver {SAVE Results to FILE}
          wm iconname .saver {SAVE RESULTS}

          set oldFocus3 [focus]

          frame .saver.top -relief raised -bd 1
          pack .saver.top -fill both
          frame .saver.bot -relief raised -bd 1
          pack .saver.bot -fill both

          label .saver.label10 -text "Please input the name of the file:"

          set filename $to_file

          set oldFocus4 [focus]

          entry .saver.name -width 20 -relief sunken -bd 2 -textvariable filename

          button .saver.doit -text "SAVE" -command "ok {$filename} {$mystuff}"
          button .saver.cancelit -text "CANCEL" -command "destroy .saver"

          pack .saver.label10 .saver.name -in .saver.top -side left -padx 2m -pady 2m -
ipadx 2m -ipady 1m
          pack .saver.doit .saver.cancelit -in .saver.bot -side left -padx 2m -pady 2m
-ipadx 2m -ipady 1m
          bind .saver.name <Return> "ok $filename {$mystuff}"
}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc ok {the_filename outstuff} {
  global filename
     set dd [open $filename w+]
     puts $dd $outstuff
     close $dd

     toplevel .ok -class OK
     wm title .ok {SAVED!!}
     wm iconname .ok OK


     frame .ok.top -relief raised -bd 1
     pack .ok.top -fill both
     frame .ok.bot -relief raised -bd 1
     pack .ok.bot -fill both

     label .ok.label100 -text " was saved correctly!"
     label .ok.msg    -text $filename
     button .ok.doit -text "OK" -command {doneit2 .ok .saver}

     pack .ok.msg .ok.label100 -in .ok.top -side left
     pack .ok.doit -in .ok.bot -side left  -pady 2m -ipadx 2m -ipady 1m

}
```

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc spier {w title text bitmap default args} {
   global button

# page 268-269

        toplevel $w -class Dialog
        wm title $w $title
        wm iconname $w {SPINET Launcher}
        frame $w.top -relief raised -bd 1
        pack $w.top -fill both
        frame $w.bot -relief raised -bd 1
        pack $w.bot -fill both

        message $w.top.msg -width 5i -text $text -font -Adobe-Times-Medium-R-Normal-*-18-
*
        pack $w.top.msg -side right -expand 1 -fill both -padx 3m -pady 3m

        if {$bitmap !="" } {
            label $w.top.bitmap -bitmap $bitmap
            pack $w.top.bitmap -side left -padx 3m -pady 3m
         }
        set i 0
        foreach but $args {
           button $w.bot.button$i -text $but -command "set button $i"
           if {$i == $default} {
              frame $w.bot.default -relief sunken -bd 1
              raise $w.bot.button$i
              pack $w.bot.default -side left -expand 1 -padx 3m -pady 3m
              pack $w.bot.button$i -in $w.bot.default -side left -padx 2m -pady 2m -ipadx
 2m -ipady 1m
              } else {
              pack $w.bot.button$i -side left -padx 2m -pady 2m -ipadx 2m -ipady 1m
              }
             incr i
           }



         if {$default >= 0} {
              bind $w <Return> "$w.bot.button$default flash; set button $default"
           }

        while {1 > 0} {
        tkwait variable button

        if {$button == 0} {
           destroy $w
           update idletasks
           exec /SPI-NET/SAB5/spin-0.96/binm/spinet
        }

        if {$button == 1} {
            destroy $w
```

```
        }

        if {$button == 2} {
           cd /SALSA
           HelpLoad HELP/spi_help "SPI-NET"
        }


}
}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc tiger {text bitmap default args} {
   global button tigerex

# page 268-269

        toplevel .tiger_launch -class Dialog
        wm title .tiger_launch {The TIGER Script Launcher}
        wm iconname .tiger_launch {TIGER Launcher}

        frame .tiger_launch.top -relief raised -bd 1
        pack .tiger_launch.top -fill both

        frame .tiger_launch.bot -relief raised -bd 1
        pack .tiger_launch.bot -fill both -side bottom

        frame .tiger_launch.mid -relief ridge -bd 3
        pack .tiger_launch.mid -fill both -side left

        message .tiger_launch.top.msg -width 5i -text $text -font -Adobe-Times-Medium-R-
Normal-*-18-*
        pack .tiger_launch.top.msg -side right -expand 1 -fill both -padx 3m -pady 3m

        if {$bitmap !="" } {
            label .tiger_launch.top.bitmap -bitmap $bitmap
            pack .tiger_launch.top.bitmap -side left -padx 3m -pady 3m
         }
        set i 0
        foreach but $args {
           if {$i == 0} {
              button .tiger_launch.mid.button$i -text $but -command "set button $i"
           } else {
              button .tiger_launch.bot.button$i -text $but -command "set button $i"
           }

           if {$i == 0} {
              pack .tiger_launch.mid.button$i -side left -padx 2m -pady 2m -ipadx 2m -ip
ady 1m
              } else {
              pack .tiger_launch.bot.button$i -side left -padx 2m -pady 2m -ipadx 2m -ip
ady 1m
              }
             incr i
           }
         if {$default >= 0} {
              bind .tiger_launch <Return> ".tiger_launch.bot.button$default flash; set bu
tton $default"
           }

        checkbutton .tiger_launch.mid.complete -text "Include Warning Explanations" -vari
able tigerex -anchor w

        pack .tiger_launch.mid.complete -in .tiger_launch.mid -side left -padx 1m -pady 0
m -ipadx 2m -ipady 1m


        set oldFocusT [focus]

        while {1 > 0} {
        tkwait variable button

        if {$button == 0} {
           cd /TIGER/tiger-2.2.3/
           if {$tigerex == 1} {exec ./tiger -E &}
           else {exec ./tiger &}
           cd /SALSA
           destroy .tiger_launch
        }

        if {$button == 1} {
            destroy .tiger_launch
            tigerview
          }

        if {$button == 2} {
            destroy .tiger_launch
            tigerclear
          }

        if {$button == 3} {
            destroy .tiger_launch
          }

        if {$button == 4} {
                cd /SALSA
                HelpLoad HELP/tiger_help "TIGER"
          }


}
}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc netview {} {
       global host people choice

        toplevel .netview -class Dialog
        wm title .netview {Current NET Status}
        wm iconname .netview {NET STATUS}
```

```
        frame .netview.btm -relief raised -bd 3
        pack .netview.btm -side bottom -padx 2m -pady 2m

        frame .netview.bottom -relief ridge -bd 3
        pack .netview.bottom -side bottom -fill both

        text .netview.results -relief sunken -width 80 -bd 2 -bg white -yscrollcommand "
.netview.scroll set"

        pack .netview.results -side left -fill both -pady 2m

        scrollbar .netview.scroll -command ".netview.results yview"

        pack .netview.scroll -side left -fill y -pady 2m

        set people {}
        set host [exec hostname]

        set choice 3

        .netview.results insert end $people

        label .netview.label -text "Net Status on:"
        entry .netview.name -width 20 -relief sunken -bd 2 -fg blue -textvariable host

        pack  .netview.label -in .netview.bottom -side left
        pack  .netview.name -in .netview.bottom -side left

        button .netview.exit -text {QUIT} -command {destroy .netview}
        button .netview.save -text {SAVE REPORT} -command {saveit $people netstat.out}

        button .netview.machine -text {Check Net Status} -command {
             .netview.results delete 1.0 end
             .netview.results insert end "\n\n\n\n                WORKING....."
             update idletasks

             if {$choice == 1} {
             set people [exec su $the_user -c "rsh $host netstat -P tcp"]
             }

             if {$choice == 2} {
             set people [exec su $the_user -c "rsh $host netstat -P udp"]
             }

             if {$choice == 3} {
             set people [exec su $the_user -c "rsh $host netstat -P tcp"]
             set people [concat $people [exec su $the_user -c "rsh $host netstat -P udp"
]]

             }

             .netview.results delete 1.0 end
             .netview.results insert end $people
         }

        radiobutton .netview.tcp -text "ACTIVE TCP Sockets" -variable choice -anchor e -
value 1
        radiobutton .netview.udp -text "ACTIVE UDP Sockets" -variable choice -anchor e -
value 2
        radiobutton .netview.both -text "TCP &  UDP Sockets" -variable choice -anchor e
-value 3

        pack .netview.machine -in .netview.bottom -side left -padx 6m
        pack .netview.tcp .netview.udp .netview.both -in .netview.bottom -side top

        pack .netview.save .netview.exit -in .netview.btm -side right -pady 2m -padx 2m

        bind .netview.name <Return> {
             .netview.results delete 1.0 end
             .netview.results insert end "\n\n\n\n                WORKING....."
             update idletasks

             if {$choice == 1} {
             set people [exec su $the_user -c "rsh $host netstat -P tcp"]
             }

             if {$choice == 2} {
             set people [exec su $the_user -c "rsh $host netstat -P udp"]
             }

             if {$choice == 3} {
             set people [exec su $the_user -c "rsh $host netstat -P tcp"]
             set people [concat $people [exec su $the_user -c "rsh $host netstat -P udp"]
]

             }

             .netview.results delete 1.0 end
             .netview.results insert end $people
        }

}


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc psview {} {
       global host people choice

        toplevel .psview -class Dialog
        wm title .psview {CURRENT PROCESSES}
        wm iconname .psview {Current Processes}

        frame .psview.btm -relief raised -bd 3
        pack .psview.btm -side bottom -padx 2m -pady 2m

        frame .psview.bottom -relief ridge -bd 3
        pack .psview.bottom -side bottom -fill both -ipadx 2m -ipady 3m

        text .psview.results -relief sunken -width 80 -bd 2 -bg white -yscrollcommand ".p
sview.scroll set"

        pack .psview.results -side left -fill both -pady 2m

        scrollbar .psview.scroll -command ".psview.results yview"

        pack .psview.scroll -side left -fill y -pady 2m

        set people {}
        set host [exec hostname]
```

```
        set choice 3

        .psview.results insert end $people

        label .psview.label -text "Current Processes on:"
        entry .psview.name -width 20 -relief sunken -bd 2 -fg blue -textvariable host

        pack  .psview.label -in .psview.bottom -side left
        pack  .psview.name -in .psview.bottom -side left

        button .psview.exit -text {QUIT} -command {destroy .psview}
        button .psview.save -text {SAVE REPORT} -command {saveit $people processes.out}

        button .psview.machine -text {Check Current Processes} -command {
             .psview.results delete 1.0 end
             .psview.results insert end "\n\n\n\n                WORKING....."
             update idletasks
             set people [exec su $the_user -c "rsh $host ps -e -o user -o fname -o comm"
]
             .psview.results delete 1.0 end
             .psview.results insert end $people
         }


        pack .psview.machine -in .psview.bottom -side left -padx 6m
        pack .psview.save .psview.exit -in .psview.btm -side right -pady 2m -padx 2m


        bind .psview.name <Return> {
             .psview.results delete 1.0 end
             .psview.results insert end "\n\n\n\n                WORKING....."
             update idletasks
             set people [exec su $the_user -c "rsh $host ps -e -o user -o fname -o comm"
 2> /dev/null]
             .psview.results delete 1.0 end
             .psview.results insert end $people
        }

}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc tigerview {} {
   global button request

# page 268-269

        toplevel .tiger_view -class Dialog
        wm title .tiger_view {The TIGER Results Viewer}
        wm iconname .tiger_view {TIGER Viewer}

        frame .tiger_view.top -relief raised -bd 1
        pack .tiger_view.top -fill both -side top

        frame .tiger_view.fname -relief raised -bd 1
        pack .tiger_view.fname -fill both -side top

        label .tiger_view.label10 -text "Please input the name of the file:"

        entry .tiger_view.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .tiger_view.label10 .tiger_view.name -in .tiger_view.fname -side left

        frame .tiger_view.pan -relief sunken -bd 1
        pack .tiger_view.pan -side bottom

        button .tiger_view.no_view -text {QUIT TIGER VIEW} -command {destroy .tiger_view}

        pack .tiger_view.no_view -in .tiger_view.pan -fill both

        set all_reports [glob /TIGER/tiger-2.2.3/security.report.*]

        regsub -all " " $all_reports "\n" format_dir

        message .tiger_view.msg -width 3i -text {Please double-click on the file that you
 wish to view} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .tiger_view.msg -in .tiger_view.top

        listbox .tiger_view.results -relief sunken -width 70 -bd 2 -bg white -yscrollcomm
and ".tiger_view.scroll set"

        pack .tiger_view.results -side left -fill both -pady 2m

        scrollbar .tiger_view.scroll -command ".tiger_view.results yview"

        pack .tiger_view.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /TIGER/tiger-2.2.3/security.report.*]] {
            .tiger_view.results insert end $i
            set request $i
        }

        bind .tiger_view.scroll <Double-Button-1> {
              .tiger_view.results yview
        }

        bind .tiger_view <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defin
ed}} {
               LoadFile $request
               destroy .tiger_view
            }
        }


        bind .tiger_view.name <Return> {
            LoadFile $request
            destroy .tiger_view
        }



    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc tigerclear {} {
   global button request

# page 268-269

        toplevel .tiger_clear -class Dialog
        wm title .tiger_clear {The TIGER Results Cleaner}
        wm iconname .tiger_clear {TIGER Cleaner}

        frame .tiger_clear.top -relief raised -bd 1
        pack .tiger_clear.top -fill both -side top

        frame .tiger_clear.fname -relief raised -bd 1
        pack .tiger_clear.fname -fill both -side top

        label .tiger_clear.label10 -text "Please input the name of the file:"

        entry .tiger_clear.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .tiger_clear.label10 .tiger_clear.name -in .tiger_clear.fname -side left

        frame .tiger_clear.pan -relief sunken -bd 1
        pack .tiger_clear.pan -side bottom

        button .tiger_clear.no_clear -text {QUIT TIGER REPORT DELETE} -command {destroy
.tiger_clear}
        pack .tiger_clear.no_clear -in .tiger_clear.pan -fill both

        set all_reports [glob /TIGER/tiger-2.2.3/security.report.*]

        regsub -all " " $all_reports "\n" format_dir

        message .tiger_clear.msg -width 3i -text {Please double-click on the file that y
ou wish to delete} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .tiger_clear.msg -in .tiger_clear.top

        listbox .tiger_clear.results -relief sunken -width 70 -bd 2 -bg white -yscrollco
mmand ".tiger_clear.scroll set"

        pack .tiger_clear.results -side left -fill x -pady 2m

        scrollbar .tiger_clear.scroll -command ".tiger_clear.results yview"

        pack .tiger_clear.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /TIGER/tiger-2.2.3/security.report.*]] {
            .tiger_clear.results insert end $i
            set request $i
        }

        bind .tiger_clear.scroll <Double-Button-1> {
              .tiger_clear.results yview
        }

        bind .tiger_clear <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defin
ed}} {
              delfile $request
              destroy .tiger_clear
            }
        }


        bind .tiger_clear.name <Return> {
            delfile $request
            destroy .tiger_clear
        }



    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc sandsview {} {
   global button request

# page 268-269

        toplevel .sands_view -class Dialog
        wm title .sands_view {The SANDS Report Viewer}
        wm iconname .sands_view {SANDS Viewer}

        frame .sands_view.top -relief raised -bd 1
        pack .sands_view.top -fill both -side top

        frame .sands_view.fname -relief raised -bd 1
        pack .sands_view.fname -fill both -side top

        label .sands_view.label10 -text "Please input the name of the file:"

        entry .sands_view.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .sands_view.label10 .sands_view.name -in .sands_view.fname -side left

        frame .sands_view.pan -relief sunken -bd 1
        pack .sands_view.pan -side bottom

        button .sands_view.no_view -text {QUIT SANDS VIEW} -command {destroy .sands_view}

        pack .sands_view.no_view -in .sands_view.pan -fill both

        set all_reports [glob /SANDS/sands.*]

        regsub -all " " $all_reports "\n" format_dir

        message .sands_view.msg -width 3i -text {Please double-click on the file that you
 wish to view} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .sands_view.msg -in .sands_view.top

        listbox .sands_view.results -relief sunken -width 70 -bd 2 -bg white -yscrollcomm
```

```
and ".sands_view.scroll set"

        pack .sands_view.results -side left -fill both -pady 2m

        scrollbar .sands_view.scroll -command ".sands_view.results yview"

        pack .sands_view.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /SANDS/sands.*]] {
            .sands_view.results insert end $i
            set request $i
        }

        bind .sands_view.scroll <Double-Button-1> {
              .sands_view.results yview
        }

        bind .sands_view <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defi
ned}} {
               CopsLoadFile $request "SANDS"
               destroy .sands_view
            }
        }


        bind .sands_view.name <Return> {
            CopsLoadFile $request "SANDS"
            destroy .sands_view
        }



    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc sandsclear {} {
   global button request

# page 268-269

        toplevel .sands_clear -class Dialog
        wm title .sands_clear {The SANDS Reports Deleter}
        wm iconname .sands_clear {SANDS Deleter}

        frame .sands_clear.top -relief raised -bd 1
        pack .sands_clear.top -fill both -side top

        frame .sands_clear.fname -relief raised -bd 1
        pack .sands_clear.fname -fill both -side top

        label .sands_clear.label10 -text "Please input the name of the file:"

        entry .sands_clear.name -width 50 -relief sunken -bd 2 -textvariable request
```

```
        pack .sands_clear.label10 .sands_clear.name -in .sands_clear.fname -side left

        frame .sands_clear.pan -relief sunken -bd 1
        pack .sands_clear.pan -side bottom

        button .sands_clear.no_clear -text {QUIT SANDS REPORT DELETE} -command {destroy .
sands_clear}
        pack .sands_clear.no_clear -in .sands_clear.pan -fill both

        set all_reports [glob /SANDS/sands.*]

        regsub -all " " $all_reports "\n" format_dir

        message .sands_clear.msg -width 3i -text {Please double-click on the file that yo
u wish to delete} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .sands_clear.msg -in .sands_clear.top

        listbox .sands_clear.results -relief sunken -width 70 -bd 2 -bg white -yscrollcom
mand ".sands_clear.scroll set"

        pack .sands_clear.results -side left -fill x -pady 2m

        scrollbar .sands_clear.scroll -command ".sands_clear.results yview"

        pack .sands_clear.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /SANDS/sands.*]] {
            .sands_clear.results insert end $i
            set request $i
        }

        bind .sands_clear.scroll <Double-Button-1> {
              .sands_clear.results yview
        }

        bind .sands_clear <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defin
ed}} {
              delfile $request
              destroy .sands_clear
            }
        }


        bind .sands_clear.name <Return> {
            delfile $request
            destroy .sands_clear
        }



    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
```

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc delfile {del_file} {

     toplevel .delfile -class dialog
     wm title .delfile {DELETED!!}
     wm iconname .delfile DELETE

     frame .delfile.top -relief raised -bd 1
     pack .delfile.top -fill both
     frame .delfile.bot -relief raised -bd 1
     pack .delfile.bot -fill both

     label .delfile.label100 -text " is about to be DELETED!"
     label .delfile.msg    -text $del_file
     button .delfile.doit -text "DELETE FILE" -command {
          exec rm $request
          destroy .delfile
         }
     button .delfile.abort -text "ABORT" -command {destroy .delfile}

     pack .delfile.msg .delfile.label100 -in .delfile.top -side left
     pack .delfile.doit .delfile.abort -in .delfile.bot -side left  -pady 2m -ipadx 2m -
ipady 1m

}
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc expview {} {
   global request c

        toplevel .expview -class Dialog
        wm title .expview {The TIGER Warnings Explainer}
        wm iconname .expview {TIGER Explainer}

        set c {}
        set request {}

        frame .expview.fname -relief raised -bd 1
        pack .expview.fname -fill both -side top

        label .expview.label10 -text "Please enter the warning:"

        entry .expview.name -width 50 -relief sunken -bd 2 -textvariable request

        frame .expview.pan -relief sunken -bd 1
        pack .expview.pan -side bottom

        pack .expview.label10 .expview.name -in .expview.fname -side left

        button .expview.no_view -text {QUIT} -command {destroy .expview}
        button .expview.one_exp -text {EXPLAIN} -command {
             .expview.results delete 1.0 end
             .expview.results insert end "\n\n\n\n      Working....."
             update idletasks
             cd /TIGER/tiger-2.2.3/
             set c [exec ./tigexp $request]
             .expview.results delete 1.0 end
             .expview.results insert end $c
             cd /SALSA
        }


        text .expview.results -relief sunken -width 80 -height 10 -bd 2 -bg black -fg whi
te -yscrollcommand ".expview.scroll set"

        pack .expview.results -side left -pady 2m
        pack .expview.one_exp .expview.no_view -in .expview.pan -fill both -side left

        scrollbar .expview.scroll -command ".expview.results yview"

        pack .expview.scroll -side left -fill y -pady 2m


        bind .expview.name <Return> {
             .expview.results delete 1.0 end
             .expview.results insert end "\n\n\n\n      Working....."
             update idletasks
             cd /TIGER/tiger-2.2.3/
             set c [exec ./tigexp $request]
             .expview.results delete 1.0 end
             .expview.results insert end $c
             cd /SALSA
        }



    }


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx


proc sepview {report_file} {
   global mytempfile c

        toplevel .sepview -class Dialog
        wm title .sepview {The COMPLETE TIGER Warnings Explainer}
        wm iconname .sepview {COMPLETE Explainer}

        set c {}
        set mytempfile $report_file

        frame .sepview.pan -relief sunken -bd 1
        pack .sepview.pan -side bottom

        button .sepview.no_view -text {QUIT} -command {destroy .sepview}

        text .sepview.results -relief sunken -width 80 -bd 2 -bg black -fg white -yscroll
command ".sepview.scroll set"

        pack .sepview.results -side left -pady 2m -fill both
```

```
        pack .sepview.no_view -in .sepview.pan -fill both -side left

        scrollbar .sepview.scroll -command ".sepview.results yview"

        pack .sepview.scroll -side left -fill y -pady 2m

        .sepview.results delete 1.0 end
        .sepview.results insert end "\n\n\n\n      Working....."
        update idletasks
        cd /TIGER/tiger-2.2.3/
        set c [exec ./tigexp -f $mytempfile]
        .sepview.results delete 1.0 end
        .sepview.results insert end $c
        cd /SALSA
        bind .sepview <Return> {
                destroy .sepview
        }


    }



#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

    proc LoadFile {tiger_file} {
        global temp_file

        toplevel .tiger_report -class Dialog
        wm title .tiger_report {The TIGER Reports}
        wm iconname .tiger_report {TIGER Reports}

        set temp_file $tiger_file

        frame .tiger_report.top -relief raised -bd 1
        pack .tiger_report.top -fill both -side top

        frame .tiger_report.pan -relief sunken -bd 1
        pack .tiger_report.pan -side bottom

        button .tiger_report.no_report -text {QUIT TIGER REPORT} -command {destroy .tige
r_report}

        button .tiger_report.one_exp -text {LOOK-UP Warnings} -command {expview}

        button .tiger_report.all_exp -text {PUT ALL Exlpanations into the Report} -comma
nd {
             .tiger_report.results delete 1.0 end
             .tiger_report.results insert end "\n\n\n\n      Working....."
             update idletasks
             set i [exec /TIGER/tiger-2.2.3/tigexp -F $temp_file]
             .tiger_report.results delete 1.0 end
             .tiger_report.results insert end $i
         }

        button .tiger_report.sep_exp -text {EXPLAIN ALL Warinings} -command {sepview $tem
p_file}


        pack .tiger_report.one_exp .tiger_report.sep_exp .tiger_report.no_report -in .tig
er_report.pan -fill both


        message .tiger_report.msg -width 5i -text {The TIGER Report you Requested.} -font
 -Adobe-Courier-bold-R-Normal-*-18-*

        pack .tiger_report.msg -in .tiger_report.top

        text .tiger_report.results -relief sunken -width 80 -height 30 -bd 2 -bg white -y
scrollcommand ".tiger_report.scroll set"

        pack .tiger_report.results -side left -fill both -pady 2m

        scrollbar .tiger_report.scroll -command ".tiger_report.results yview"

        pack .tiger_report.scroll -side left -fill y -pady 2m

        .tiger_report.results delete 1.0 end
        set f [open $tiger_file]

        while {![eof $f]} {
            .tiger_report.results insert end [read $f 200]
        }
        close $f
    }


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc loglaunch {text default args}  {
    global button

    toplevel .logs_launch -class Dialog
    wm title .logs_launch {The LOG Cutter}
    wm iconname .logs_launch {LOG LAUNCHER}

    frame .logs_launch.top -relief raised -bd 1
    pack .logs_launch.top -fill both

    frame .logs_launch.bot -relief raised -bd 2
    pack .logs_launch.bot -fill both -side left

    frame .logs_launch.bot.left  -relief raised -bd 2
    pack .logs_launch.bot.left -fill both -side left -in .logs_launch.bot

    frame .logs_launch.bot.mid -relief raised -bd 2
    pack .logs_launch.bot.mid -fill both -side left

    frame .logs_launch.bot.right -relief raised -bd 1
    pack .logs_launch.bot.right -fill both -side right -in .logs_launch.bot -anchor e
```

```
    message .logs_launch.top.msg -width 5i -text $text -font -Adobe-Times-Medium-R-Norma
l-*-18-*
    pack .logs_launch.top.msg -side right -expand 1 -fill both -padx 3m -pady 3m
    set i 0
    foreach but $args {
        button .logs_launch.bot.button$i -text $but -command "set button $i"
        if {$i == $default} {
            frame .logs_launch.bot.default -relief sunken -bd 1
            raise .logs_launch.bot.button$i
            pack .logs_launch.bot.default -side left -expand 1 -padx 3m -pady 3m
            pack .logs_launch.bot.button$i -in .logs_launch.bot.default -side left -padx
 2m -pady 2m -ipadx 2m -ipady 1m
        }

        if {$i < 2} {
            pack .logs_launch.bot.button$i -side top -in .logs_launch.bot.left -padx 2m
-pady 2m -ipadx 2m -ipady 1m
        }

        if {($i > 1) && ($i < 5)} {
            pack .logs_launch.bot.button$i -side top -in .logs_launch.bot.mid -padx 2m -
pady 2m -ipadx 2m -ipady 1m
        }

        if {$i > 4} {
            pack .logs_launch.bot.button$i -side right -in .logs_launch.bot.right -padx
2m -pady 2m -ipadx 2m -ipady 1m
        }

        incr i
    }


    while {1 > 0} {
        tkwait variable button

        if {$button == 0} {
            whoview
            destroy .logs_launch
        }

        if {$button == 1} {
            destroy .logs_launch
            lastview
        }

        if {$button == 2} {
            destroy .logs_launch
            netview
        }

        if {$button == 3} {
            destroy .logs_launch
            psview
        }

        if {$button == 4} {
            destroy .logs_launch
            findchanges
        }

        if {$button == 5} {
            destroy .logs_launch
        }
        if {$button == 6} {
           cd /SALSA
           HelpLoad HELP/log_help "LOG CUTTER"
        }


    }
}



#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc cops {text bitmap default args} {
   global button

# page 268-269

        toplevel .cops_launch -class Dialog
        wm title .cops_launch {Computer Oracle and Password System}
        wm iconname .cops_launch {COPS Launcher}

        frame .cops_launch.top -relief raised -bd 1
        pack .cops_launch.top -fill both
        frame .cops_launch.bot -relief raised -bd 1
        pack .cops_launch.bot -fill both

        message .cops_launch.top.msg -width 5i -text $text -font -Adobe-Times-Medium-R-No
rmal-*-18-*
        pack .cops_launch.top.msg -side right -expand 1 -fill both -padx 3m -pady 3m

        if {$bitmap !=""} {
            label .cops_launch.top.bitmap -bitmap $bitmap
            pack .cops_launch.top.bitmap -side left -padx 3m -pady 3m
         }
        set i 0
        foreach but $args {
           button .cops_launch.bot.button$i -text $but -command "set button $i"
           if {$i == $default} {
              frame .cops_launch.bot.default -relief sunken -bd 1
              raise .cops_launch.bot.button$i
              pack .cops_launch.bot.default -side left -expand 1 -padx 3m -pady 3m
              pack .cops_launch.bot.button$i -in .cops_launch.bot.default -side left -pad
x 2m -pady 2m -ipadx 2m -ipady 1m
              } else {
              pack .cops_launch.bot.button$i -side left -padx 2m -pady 2m -ipadx 2m -ipad
y 1m
              }
             incr i
           }



         if {$default >= 0} {
              bind .cops_launch <Return> ".cops_launch.bot.button$default flash; set butt
on $default"
```

```
         }

        set oldFocusT [focus]

        while {1 > 0} {
        tkwait variable button

        if {$button == 0} {
           cd /COPS/cops_104/
           exec ./cops -v -s . -b cops.err &
           destroy .cops_launch
           cd /SALSA
        }

        if {$button == 1} {
            destroy .cops_launch
            copsview
          }

        if {$button == 2} {
            destroy .cops_launch
            copsclear
          }

        if {$button == 3} {
            destroy .cops_launch
          }

        if {$button == 4} {
            cd /COPS/cops_104/
            HelpLoad README1 "COPS"
            cd /SALSA
          }


}
}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc copsview {} {
   global button request

# page 268-269

        toplevel .cops_view -class Dialog
        wm title .cops_view {The COPS Report Viewer}
        wm iconname .cops_view {COPS Viewer}

        frame .cops_view.top -relief raised -bd 1
        pack .cops_view.top -fill both -side top

        frame .cops_view.fname -relief raised -bd 1
        pack .cops_view.fname -fill both -side top

        label .cops_view.label10 -text "Please input the name of the file:"

        entry .cops_view.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .cops_view.label10 .cops_view.name -in .cops_view.fname -side left

        frame .cops_view.pan -relief sunken -bd 1
        pack .cops_view.pan -side bottom

        button .cops_view.no_view -text {QUIT COPS VIEW} -command {destroy .cops_view}
        pack .cops_view.no_view -in .cops_view.pan -fill both

#       set all_reports [glob /COPS/cops_104/result.*]

#       regsub -all " " $all_reports "\n" format_dir

        message .cops_view.msg -width 3i -text {Please double-click on the file that you
wish to view} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .cops_view.msg -in .cops_view.top

        listbox .cops_view.results -relief sunken -width 50 -bd 2 -bg white -yscrollcomma
nd ".cops_view.scroll set"

        pack .cops_view.results -side left -fill x -pady 2m

        scrollbar .cops_view.scroll -command ".cops_view.results yview"

        pack .cops_view.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /COPS/cops_104/result.* /COPS/cops_104/hansolo/*]] {
            .cops_view.results insert end $i
            set request $i
        }

        bind .cops_view.scroll <Double-Button-1> {
              .cops_view.results yview
        }

        bind .cops_view <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defin
ed}} {
              CopsLoadFile $request COPS
              destroy .cops_view
            }
        }

        bind .cops_view.name <Return> {
            CopsLoadFile $request COPS
            destroy .cops_view
        }



    }


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
xx

proc copsclear {} {
   global button request

# page 268-269

        toplevel .cops_clear -class Dialog
        wm title .cops_clear {The COPS Results Cleaner}
        wm iconname .cops_clear {COPS Cleaner}

        frame .cops_clear.top -relief raised -bd 1
        pack .cops_clear.top -fill both -side top

        frame .cops_clear.fname -relief raised -bd 1
        pack .cops_clear.fname -fill both -side top

        label .cops_clear.label10 -text "Please input the name of the file:"

        entry .cops_clear.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .cops_clear.label10 .cops_clear.name -in .cops_clear.fname -side left

        frame .cops_clear.pan -relief sunken -bd 1
        pack .cops_clear.pan -side bottom

        button .cops_clear.no_clear -text {QUIT COPS REPORT DELETE} -command {destroy .c
ops_clear}
        pack .cops_clear.no_clear -in .cops_clear.pan -fill both

#       set all_reports [glob /COPS/cops_104/result.*]

#       regsub -all " " $all_reports "\n" format_dir

        message .cops_clear.msg -width 3i -text {Please double-click on the file that yo
u wish to delete} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .cops_clear.msg -in .cops_clear.top

        listbox .cops_clear.results -relief sunken -width 70 -bd 2 -bg white -yscrollcom
mand ".cops_clear.scroll set"

        pack .cops_clear.results -side left -fill x -pady 2m

        scrollbar .cops_clear.scroll -command ".cops_clear.results yview"

        pack .cops_clear.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /COPS/cops_104/result.* /COPS/cops_104/hansolo/*]] {
            .cops_clear.results insert end $i
            set request $i
        }


        bind .cops_clear.scroll <Double-Button-1> {
             .cops_clear.results yview
        }

        bind .cops_clear <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defi
ned}} {
              delfile $request
              destroy .cops_clear
            }
        }


        bind .cops_clear.name <Return> {
            delfile $request
            destroy .cops_clear
        }



    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

    proc CopsLoadFile {cops_file temp_name} {

        global caller

        set caller $temp_name

        toplevel .cops_report -class Dialog
        wm title .cops_report "The $caller Reports"
        wm iconname .cops_report "$caller Reports"

        frame .cops_report.top -relief raised -bd 1
        pack .cops_report.top -fill both -side top

        frame .cops_report.pan -relief sunken -bd 1
        pack .cops_report.pan -side bottom

        button .cops_report.no_report -text "QUIT $caller REPORT" -command {destroy .cops
_report}
        pack .cops_report.no_report -in .cops_report.pan -fill both


        message .cops_report.msg -width 5i -text "The $caller Report you Requested." -fon
t -Adobe-Courier-bold-R-Normal-*-18-*

        pack .cops_report.msg -in .cops_report.top

        text .cops_report.results -relief sunken -width 100 -height 30 -bd 2 -bg white -y
scrollcommand ".cops_report.scroll set"

        pack .cops_report.results -side left -fill both -pady 2m

        scrollbar .cops_report.scroll -command ".cops_report.results yview"

        pack .cops_report.scroll -side left -fill y -pady 2m

        .cops_report.results delete 1.0 end
        set f [open $cops_file]

        while {![eof $f]} {
            .cops_report.results insert end [read $f 200]
        }
```

```
        close $f
    }


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
proc password {} {
   global the_user his_password

    toplevel .password -class Dialog
    wm title .password {Please LOGIN}
    wm iconname .password {LOGIN}

    update idletasks
    raise .password .
    update idletasks

    frame .password.top
    frame .password.bottom

    set the_user dougd
        # [exec whoami]

    pack .password.top .password.bottom -side top -padx 1m -pady 1m

    label .password.username -text "System Username: "
    label .password.password -text "SALSA Password:  "

    entry .password.the_name -width 20 -relief sunken -bd 2 -textvariable the_user
    entry .password.the_word -width 20 -bg gray -fg gray -relief sunken -bd 2 -textvaria
ble his_password


    pack .password.username .password.the_name -in .password.top -side left
    pack .password.password .password.the_word -in .password.bottom -side left


    bind .password <Return> {
        cd /SALSA
        set checker [exec ./guard $the_user]
        if {$checker == $his_password} {
            set go doug
            destroy .password
        }

    }


    tkwait variable go

}

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc admin {} {
       global choice tempuser temppass newpass

        toplevel .admin -class Dialog
        wm title .admin {UPDATE PASSWORD FILE}
        wm iconname .admin {PASSWORD}

        frame .admin.btm -relief raised -bd 3
        pack .admin.btm -side bottom -padx 2m -pady 2m

        frame .admin.bottom -relief ridge -bd 3
        pack .admin.bottom -side bottom -fill both

        frame .admin.top -relief ridge -bd 3
        pack .admin.top -side bottom -fill both

        set choice 1

        set tempuser {}
        set temppass {}

        label .admin.warning -fg red -text "USERNAME ALREADY EXISTS!"
        label .admin.warning2 -fg red -text "USERNAME DOES NOT EXIST!"

        label .admin.label -text "System Username: "
        entry .admin.name -width 20 -relief sunken -bd 2 -textvariable tempuser

        label .admin.plabel -text "SALSA Password:  "
        entry .admin.pass -width 20 -relief sunken -bd 2 -textvariable temppass

        label .admin.nplabel -text "NEW SALSA Password:  "
        entry .admin.npass -width 20 -relief sunken -bd 2 -textvariable newpass

        pack  .admin.label -in .admin.top -side left
        pack  .admin.name -in .admin.top -side left

        pack  .admin.plabel -in .admin.bottom -side left
        pack  .admin.pass -in .admin.bottom -side left

        button .admin.exit -text {FORGET CHANGE} -bg red -fg yellow -command {destroy .ad
min}

        button .admin.help -text {HELP} -command {
               cd /SALSA
               HelpLoad HELP/admin_help "PASSWORD ADMINISTRATION"
        }

        button .admin.doit -text {MAKE CHANGE} -bg green -command {

             if {$choice == 1} {
                 cd /SALSA
                 catch {exec grep $tempuser ".  "} checker
                 if {$checker != "child process exited abnormally"} {
                     pack .admin.warning  -in .admin.bottom -side bottom
                 }

                 if {$checker == "child process exited abnormally"} {
                      cd /SALSA
                      exec echo $tempuser $temppass >> ".  "
                      destroy .admin
                 }
```

```
            }

            if {$choice == 2} {
                cd /SALSA
                exec grep -v "$tempuser $temppass" ".  " > .out
                exec mv .out ".  "
                destroy .admin
            }

            if {$choice == 3} {
                cd /SALSA
                catch {exec grep $tempuser ".  "} checker
                if {$checker == "child process exited abnormally"} {
                    pack .admin.warning2  -in .admin.bottom -side bottom
                }

                if {$checker != "child process exited abnormally"} {
                     cd /SALSA
                     exec grep -v "$tempuser" ".  " > .out
                     exec mv .out ".  "
                     exec echo $tempuser $temppass >> ".  "
                     destroy .admin

                }
            }

        }

        radiobutton .admin.cr -text "CREATE USER" -variable choice -anchor e -value 1
        radiobutton .admin.de -text "DELETE USER" -variable choice -anchor e -value 2
        radiobutton .admin.ch -text "CHANGE USER" -variable choice -anchor e -value 3

        pack .admin.cr .admin.ch .admin.de -in .admin.bottom -side top
        pack .admin.doit .admin.exit .admin.help -in .admin.btm -side left -pady 2m -pad
x 2m


}
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc gosatanhelp {} {
   global button request f

# page 268-269

        toplevel .gosatanhelp -class Dialog
        wm title .gosatanhelp {SATAN HELP}
        wm iconname .gosatanhelp {SATAN HELP}

        frame .gosatanhelp.top -relief raised -bd 1
        pack .gosatanhelp.top -fill both -side top

        frame .gosatanhelp.pan -relief sunken -bd 1
        pack .gosatanhelp.pan -side bottom

        button .gosatanhelp.morehelp -text {MORE SATAN HELP} -command {
             .gosatanhelp.results delete 1.0 end
             .gosatanhelp.results insert end "\n\n\n\n       This will take a moment (ab
out 50-60 seconds)....."
             update idletasks
             exec su $the_user -c "/usr/local/bin/netscape /SATAN/satan-1.1.1/html/satan.
html" &
             after 50000
             destroy .gosatanhelp
           }
        button .gosatanhelp.no_view -text {QUIT SATAN HELP} -command {destroy .gosatanhel
p}
        pack .gosatanhelp.morehelp .gosatanhelp.no_view -in .gosatanhelp.pan -fill both -
side left

        message .gosatanhelp.msg -width 4i -text {This is the SATAN man page.} -font -Ado
be-Courier-bold-R-Normal-*-18-*

        pack .gosatanhelp.msg -in .gosatanhelp.top

        text .gosatanhelp.results -relief sunken -width 105 -height 30 -bd 2 -bg black -f
g  white -yscrollcommand ".gosatanhelp.scroll set"

        pack .gosatanhelp.results -side left -fill both -pady 2m

        scrollbar .gosatanhelp.scroll -command ".gosatanhelp.results yview"

        pack .gosatanhelp.scroll -side left -fill y -pady 2m

        .gosatanhelp.results delete 1.0 end

        set f [exec troff -man -af /SATAN/satan-1.1.1/satan.8]
       .gosatanhelp.results insert end $f

    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

 proc satanfacts {} {
   global button request machine

# page 268-269

        toplevel .satanfacts -class Dialog
        wm title .satanfacts {SATAN VIEW}
        wm iconname .satanfacts {SATAN VIEW}

        set machine all

        frame .satanfacts.top -relief raised -bd 1
        pack .satanfacts.top -fill both -side top

        frame .satanfacts.pan -relief sunken -bd 1
        pack .satanfacts.pan -side bottom -pady 3m

        frame .satanfacts.input
        pack .satanfacts.input -side bottom -pady 3m

        label .satanfacts.label -text "Machine to get facts on: "
```

```
        entry .satanfacts.entry -textvariable machine -fg blue

        pack .satanfacts.label .satanfacts.entry -in .satanfacts.input -side left

        button .satanfacts.morehelp -text "CHECK FACTS" -command {
             .satanfacts.results delete 1.0 end
             .satanfacts.results insert end "\n\n\n\n\n         WORKING...."
             update idletasks
             if {($machine == "All") || ($machine == "ALL") ||($machine == "all")} {
                  cd /SALSA
                  set facts [exec ./runner]
             } else {
                  cd /SALSA
                  set facts [exec ./runner | grep $machine]
             }
             .satanfacts.results delete 1.0 end
             .satanfacts.results insert end $facts
           }


        button .satanfacts.no_view -text {QUIT SATAN FACTS} -command {destroy .satanfact
s}

        button .satanfacts.help -text {HELP} -command {
                cd /SALSA
                HelpLoad HELP/satan_facts_help "SATAN facts"
        }

        pack .satanfacts.morehelp .satanfacts.no_view .satanfacts.help -in .satanfacts.p
an -fill both -side left

        message .satanfacts.msg -width 4i -text "Known Facts." -font -Adobe-Courier-bold
-R-Normal--*-18-*

        pack .satanfacts.msg -in .satanfacts.top

        text .satanfacts.results -relief sunken -width 80 -height 30 -bd 2 -bg white -ys
crollcommand ".satanfacts.scroll set"

        pack .satanfacts.results -side left -fill both -pady 2m

        scrollbar .satanfacts.scroll -command ".satanfacts.results yview"

        pack .satanfacts.scroll -side left -fill y -pady 2m

        .satanfacts.results delete 1.0 end

        cd /SALSA
        set facts [exec ./runner]
        update idletasks

       .satanfacts.results insert end $facts

    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
proc gosatan {} {
```

```
    global target level

     toplevel .gosatan -class Dialog
     wm title .gosatan {SATAN TARGET SELECTION}
     wm iconname .gosatan {SATAN TARGET}

     frame .gosatan.top -relief ridge -bd 3
     pack .gosatan.top -side top -fill both -pady 4m

     set target {hansolo.cs.tamu.edu}
     set level 0

     frame .gosatan.bottom
     pack .gosatan.bottom -side top -fill both


     frame .gosatan.pan -relief raised -bd 2
     pack .gosatan.pan -side top -fill both -pady 1m

     label .gosatan.label -text "Input the target name: "

     label .gosatan.label3 -textvariable  level -fg blue -relief sunken -width 3
     entry .gosatan.entry -textvariable target -fg blue

     menubutton .gosatan.alevel -text "Attack Level" -menu .gosatan.alevel.menu

     menu .gosatan.alevel.menu

     .gosatan.alevel.menu add command -label {0} -command {
         set level 0
     }

     .gosatan.alevel.menu add command -label {1} -command {
         set level 1
     }

     .gosatan.alevel.menu add command -label {2} -command {
         set level 2
     }

     tk_menuBar .gosatan.bottom .gosatan.alevel

     focus .gosatan.bottom

     pack .gosatan.label .gosatan.entry -in .gosatan.top -side left -padx 1m -pady 2m
     pack .gosatan.label3 .gosatan.alevel -in .gosatan.top -side right -padx 1m -pady
2m


#     pack .gosatan.alevel -side right -padx 1m


     button .gosatan.check -text "CHECK TARGET" -fg red -command {
         cd /SATAN/satan-1.1.1
         destroy .gosatan.top
         label .gosatan.label2 -text "PLEASE WAIT WHILE SATAN CHECKS:  $target.\n
(It may take a moment)" -fg red -bg yellow
         pack .gosatan.label2 -in .gosatan.bottom -padx 2m -pady 2m
         update idletasks
         exec ./satan -a $level $target
         destroy .gosatan
         cd /SALSA
      }
```

```
        button .gosatan.exit -text "EXIT" -command {destroy .gosatan}

        button .gosatan.help -text "HELP" -command {
                cd /SALSA
                HelpLoad HELP/satan_target_help "SATAN Target\nSelection"
        }
        pack .gosatan.check .gosatan.exit .gosatan.help -in .gosatan.pan -fill both -sid
e left -padx 3m

        }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx

proc satan {} {
       global doug

        toplevel .satan -class Dialog
        wm title .satan {Security Administrator Tool for Analyzing Networks}
        wm iconname .satan {SATAN}

        frame .satan.top -relief ridge -bd 3
        pack .satan.top -side top -fill both

        frame .satan.bottom
        pack .satan.bottom -side top -fill both

        message .satan.top.msg -width 5i -text {SATAN is a tool developed by Dan Farmer
and Wieste Venema.  It is a network port scanner.  Because of the nature of this tool, i
t is only partially implemented in SALSA.} -font -Adobe-Times-Medium-R-Normal-*-18-*

       pack .satan.top.msg -side right -in .satan.top -expand 1 -fill both -padx 3m -pad
y 3m

        button .satan.launch -text {LAUNCH SATAN} -command {
            gosatan
            destroy .satan
        }
        button .satan.view -text {VIEW SATAN FACTS} -command {
            satanfacts
            destroy .satan
        }
        button .satan.interactive -text {HELP} -command {
            gosatanhelp
            destroy .satan
        }
        button .satan.exit -text {QUIT} -command {destroy .satan}


        pack .satan.launch .satan.view .satan.interactive .satan.exit -fill both -in .sa
tan.bottom -side left -padx 2m -pady 2m

}
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

    proc HelpLoad {help_file help_info} {


        toplevel .help_$help_file -class Dialog
        wm title .help_$help_file {SALSA HELP}
        wm iconname .help_$help_file {HELP}

        frame .help_$help_file.top -relief raised -bd 1
        pack .help_$help_file.top -fill both -side top

        frame .help_$help_file.pan -relief sunken -bd 1
        pack .help_$help_file.pan -side bottom

        button .help_$help_file.no_report -text {QUIT HELP} -command "destroy .help_$help
_file"
        pack .help_$help_file.no_report -in .help_$help_file.pan -fill both

        message .help_$help_file.msg -width 5i -text "Help About $help_info" -font -adobe
-times-bold-i-normal--24-240-75-75-p-128-iso8859-1

        pack .help_$help_file.msg -in .help_$help_file.top

        text .help_$help_file.results -relief sunken -width 80 -height 20 -bd 2 -bg black
 -fg white -font -adobe-courier-bold-r-normal--14-140-75-75-m-90-iso8859-1 -yscrollcomman
d ".help_$help_file.scroll set"

        pack .help_$help_file.results -side left -fill both -pady 2m

        scrollbar .help_$help_file.scroll -command ".help_$help_file.results yview"

        pack .help_$help_file.scroll -side left -fill y -pady 2m

        .help_$help_file.results delete 1.0 end
        set f [open $help_file]

        while {![eof $f]} {
            .help_$help_file.results insert end [read $f 200]
        }
        close $f
    }


#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc crackview {} {
   global button request

# page 268-269

        toplevel .crack_view -class Dialog
        wm title .crack_view {The CRACK Results Viewer}
        wm iconname .crack_view {CRACK Viewer}
```

```
        frame .crack_view.top -relief raised -bd 1
        pack .crack_view.top -fill both -side top

        frame .crack_view.fname -relief raised -bd 1
        pack .crack_view.fname -fill both -side top

        label .crack_view.label10 -text "Please input the name of the file:"

        entry .crack_view.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .crack_view.label10 .crack_view.name -in .crack_view.fname -side left

        frame .crack_view.pan -relief sunken -bd 1
        pack .crack_view.pan -side bottom

        button .crack_view.no_view -text {QUIT CRACK VIEW} -command {destroy .crack_view
}

        pack .crack_view.no_view -in .crack_view.pan -fill both

        set all_reports [glob /CRACK/out.*]

        regsub -all " " $all_reports "\n" format_dir

        message .crack_view.msg -width 3i -text {Please double-click on the file that yo
u wish to view} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .crack_view.msg -in .crack_view.top

        listbox .crack_view.results -relief sunken -width 70 -bd 2 -bg white -yscrollcom
mand ".crack_view.scroll set"

        pack .crack_view.results -side left -fill both -pady 2m

        scrollbar .crack_view.scroll -command ".crack_view.results yview"

        pack .crack_view.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /CRACK/out.*]] {
            .crack_view.results insert end $i
            set request $i
        }

        bind .crack_view.scroll <Double-Button-1> {
              .crack_view.results yview
        }

        bind .crack_view <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defi
ned}} {
               CopsLoadFile $request CRACK
               destroy .crack_view
            }
        }


        bind .crack_view.name <Return> {
            LoadFile $request CRACK
            destroy .crack_view
        }
    }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc crackclear {} {
   global button request

# page 268-269

        toplevel .crack_clear -class Dialog
        wm title .crack_clear {The CRACK Results Cleaner}
        wm iconname .crack_clear {CRACK Cleaner}

        frame .crack_clear.top -relief raised -bd 1
        pack .crack_clear.top -fill both -side top

        frame .crack_clear.fname -relief raised -bd 1
        pack .crack_clear.fname -fill both -side top

        label .crack_clear.label10 -text "Please input the name of the file:"

        entry .crack_clear.name -width 50 -relief sunken -bd 2 -textvariable request

        pack .crack_clear.label10 .crack_clear.name -in .crack_clear.fname -side left

        frame .crack_clear.pan -relief sunken -bd 1
        pack .crack_clear.pan -side bottom

        button .crack_clear.no_clear -text {QUIT CRACK REPORT DELETE} -command {destroy .
crack_clear}
        pack .crack_clear.no_clear -in .crack_clear.pan -fill both

        set all_reports [glob /CRACK/out.*]

        regsub -all " " $all_reports "\n" format_dir

        message .crack_clear.msg -width 3i -text {Please double-click on the file that yo
u wish to delete} -font -Adobe-Courier-bold-R-Normal-*-18-*

        pack .crack_clear.msg -in .crack_clear.top

        listbox .crack_clear.results -relief sunken -width 70 -bd 2 -bg white -yscrollcom
mand ".crack_clear.scroll set"

        pack .crack_clear.results -side left -fill x -pady 2m

        scrollbar .crack_clear.scroll -command ".crack_clear.results yview"

        pack .crack_clear.scroll -side left -fill y -pady 2m

        foreach i [lsort [glob /CRACK/out.*]] {
            .crack_clear.results insert end $i
            set request $i
        }

        bind .crack_clear.scroll <Double-Button-1> {
```

```
              .crack_clear.results yview
        }

        bind .crack_clear <Double-Button-1> {
            catch {set request [selection get]} error_msg
            update idletasks
            if {$error_msg != {PRIMARY selection doesn't exist or form "STRING" not defi
ned}} {
              delfile $request
              destroy .crack_clear
            }
        }


        bind .crack_clear.name <Return> {
            delfile $request
            destroy .crack_clear
        }



    }



#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
proc gocrack {} {
       global target

        toplevel .gocrack -class Dialog
        wm title .gocrack {PICK THE FILE to CRACK}
        wm iconname .gocrack {CRACK TARGET}

        frame .gocrack.top -relief ridge -bd 3
        pack .gocrack.top -side top -fill both -pady 4m

        set target {/etc/passwd}


        frame .gocrack.bottom
        pack .gocrack.bottom -side top -fill both


        frame .gocrack.pan -relief raised -bd 2
        pack .gocrack.pan -side top -fill both -pady 1m

        label .gocrack.label -text "Input the password file name: "

        entry .gocrack.entry -textvariable target -fg blue

        pack .gocrack.label .gocrack.entry -in .gocrack.top -side left -padx 1m -pady 2m

        button .gocrack.check -text "CRACK IT" -fg white -bg black -command {
            cd /CRACK/
            destroy .gocrack
            update idletasks
            exec ./Crack $target > /dev/null &
            cd /SALSA
         }

        button .gocrack.exit -text "EXIT" -command {destroy .gocrack}

        pack .gocrack.check .gocrack.exit -in .gocrack.pan -fill both -side left -padx 2m

        }

#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx

proc crack {} {
       global doug

        toplevel .crack -class Dialog
        wm title .crack {CRACK}
        wm iconname .crack {CRACK}

        frame .crack.top -relief ridge -bd 3
        pack .crack.top -side top -fill both

        frame .crack.bottom
        pack .crack.bottom -side top -fill both

        message .crack.top.msg -width 5i -text {CRACK is a tool developed by Alec D. E. M
uffett.  It is designed to find standard Unix eight-charachter, DES encrypted passwords,
using standard guessing techniques.} -font -Adobe-Times-Medium-R-Normal-*-18-*

       pack .crack.top.msg -side right -in .crack.top -expand 1 -fill both -padx 3m -pady
 3m

        button .crack.launch -text {RUN CRACK} -command {
            gocrack
            destroy .crack
        }
        button .crack.view -text {VIEW CRACKED PASSWORDS} -command {
            crackview
            destroy .crack
        }
        button .crack.clear -text {DELETE CRACKED PASSWORDS} -command {
            crackclear
            destroy .crack
        }
        button .crack.interactive -text {HELP} -command {
            cd /SALSA
            HelpLoad HELP/crack_help "CRACK"
        }
        button .crack.exit -text {QUIT} -command {destroy .crack}


        pack .crack.launch .crack.view .crack.clear .crack.interactive .crack.exit -fill
both -in .crack.bottom -side left -padx 2m -pady 2m

}
```

```
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx
#xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxx



wm title . {(S)ECURITY (A)PPLICATION (L)AUNCHER for (S)YSTEM (A)DMINISTRATORS}
wm iconname . {SALSA}

password


frame .top -relief ridge -borderwidth 1m
frame .left -relief ridge -borderwidth 1m
frame .right
frame .bottom -relief raised


message .title1 -text "This is the system System Administrator Software" -font -adobe-he
lvetica-medium-o-normal--22-0-75-75-p-0-iso8859-1 -width 5i
message .title2 -text "Suite Developed at Texas A&M, Please choose one " -font -adobe-he
lvetica-medium-o-normal--22-0-75-75-p-0-iso8859-1 -width 5i
message .title3 -text "of the following options to get information of  " -font -adobe-he
lvetica-medium-o-normal--22-0-75-75-p-0-iso8859-1 -width 5i
message .title4 -text "the software and then execute it.             " -font -adobe-he
lvetica-medium-o-normal--22-0-75-75-p-0-iso8859-1 -width 5i

label .label2 -foreground blue -text "SCANNING AND DIAGNOSTIC TOOLS"
label .label3 -foreground blue -text "INTRUSION DETECTION TOOLS"
label .labelx -foreground blue -text "SALSA ADMINISTRATION HELP"

button .crack_but -foreground white -background black -text CRACK -command {crack}

button .sands -foreground purple -background white -text SANDS -command {sands {SANDS wa
s developed at Texas A&M University to help System Administrators by running a standard
security checklist published by AUCERT and reporting the results.}  {} -1 {Launch SANDS}
 {VIEW OLD SANDS Reports} {DELETE OLD SANDS Reports} {CANCEL} {HELP}}

button .cops -foreground  white -background blue -text COPS  -command {cops {COPS was de
veloped by Dan Farmer.  It is a tool similar to TIGER and SANDS.  It checks password sec
urity and several other common known vulnerabilities.}  {} -1 {Launch COPS} {View COPS R
eports} {DELETE OLD COPS Reports} {CANCEL} {HELP}}

button .satan_but -foreground red -background yellow -text SATAN -command {satan}

button .tiger -foreground yellow -background red -text TIGER -command {tiger {TIGER was
developed at Texas A&M University.  It runs extensive set of scripts that check the secu
rity of the system against many well known exploits.  It is known for generating copious
 reports.}  {} -1 {Launch TIGER} {View TIGER Reports} {DELETE OLD TIGER Reports} {CANCEL
} {HELP}}

button .white -foreground blue -background white -text WHITE -command {dialog .d {System
 And Network Diagnotic Software} {SANDS was developed at Texas A&M University to help Sy
stem Administrators by running a standard security checklist published by AUCERT and rep
orting the results.}  {} -1 {Execute WHITE} {CANCEL} {HELP}}

button .spi -foreground yellow -background purple -text SPINET -command {spier .d {Secur
ity Profile Inspector} {SPINET was developed by the Department of Energy.  It runs vario
us checks from a command host and all of the data is centralized from remotes hosts.}  {
} -1 {Launch SPINET} {CANCEL} {HELP}}

button .cron -foreground black -background green -text {PERIOD PLANNER} -command {crontoo
l}

button .adminer -foreground blue -background orange -text {PASSWORD ADMIN} -command {admi
n}

button .logs -foreground  white -background black -text "THE LOG CUTTERS" -command {logla
unch {This tool was developed in connection with SANDS at Texas A&M.  It is used to give
a system administrator to view easily view current and past logins on a variety of machin
es.} -1 {Current Logins} {Past Logins} {Net Status} {Current Processes} {Changed or Acces
sed Files} {CANCEL} {HELP}}

button .ipwat -foreground black -background green -text "IP WATCHER" -command {dialog .d
{System And Network Diagnotic Software} {SANDS was developed at Texas A&M University to h
elp System Administrators by running a standard security checklist published by AUCERT an
d reporting the results.}  {} -1 {Execute IP WATCHER} {CANCEL} {HELP}}

button .twire -foreground yellow -background orange -text TRIPWIRE -command {dialog .d {S
ystem And Network Diagnotic Software} {SANDS was developed at Texas A&M University to hel
p System Administrators by running a standard security checklist published by AUCERT and
reporting the results.}  {} -1 {Execute TRIPWIRE} {CANCEL} {HELP}}

button .exit -relief groove -text "EXIT SALSA" -command exit
button .about -relief groove -text "ABOUT SALSA" -command {
      cd /SALSA
      HelpLoad HELP/salsa_help "SALSA\nSecurity Application Launcher for System Administr
ators"
}
pack .top -side top -pady 4m
pack .left -side left -fill x -padx 10m -pady 3m
pack .right -side right -fill x -padx 10m -pady 3m

frame .right.top -relief ridge -bd 1m
frame .right.bottom -relief ridge -bd 1m

pack .right.top -in .right -side top -fill x -pady 2m
pack .right.bottom -in .right -side bottom -fill x -pady 2m

pack .bottom -side bottom -fill both -pady 4m -padx 3m

pack .title1 .title2 .title3 .title4 -in .top -side top -pady 2m
pack .label2 .sands .tiger .cops .crack_but .satan_but  -in .left -side  top  -padx 2m -p
ady 2m
pack .label3 .spi .logs -in .right.top -side top  -padx 2m -pady 2m
# .white .ipwat .twire
pack .labelx .cron .adminer -in .right.bottom -side top -padx 2m -pady 2m
pack .about .exit -in .bottom -side left -padx 4m -pady 4m
```

```
#include <stdio.h>
#include <string.h>
#include<errno.h>
#include<sys/time.h>
#include <unistd.h>
#include <fcntl.h>
#include <string.h>
#include <stdlib.h>

#define ON 1
#define OFF 0

char input[5000], input2[5000], input3[5000], buffer[5000], buffer2[5000], tempstring[5000];
char commandstring[100];
int port, xwind, forward, file_perms, equiv, rhoster, ngroup, complete, serv, rhoster_f, grower, majdomo = OFF;
int       input_fd, input_fd2;
int i, j, k, l, m, n, o, p, plus_flag;
int r_commands = 0;


/*****************************************************************/
/*  M A I N  P R O G R A M                                       */
/*****************************************************************/

main (int argc, char *argv[])
{
  bzero (commandstring, sizeof (commandstring));
  m = 1;
  while (m < argc)
  {
  strcat(commandstring,argv[m]);
  m++;
  }


i = 0;

while (i < strlen(commandstring))
  {
  switch(commandstring[i])
    {
     case 'P': port = ON; break;
     case 'S': serv = ON; break;
     case 'E': equiv = ON; break;
     case 'B': file_perms = ON; break;
     case 'F': forward = ON; break;
     case 'X': xwind = ON; break;
     case 'V': rhoster_f = ON; break;
     case 'R': rhoster = ON; break;
     case 'N': ngroup = ON; break;
     case 'M': majdomo = ON; break;
     case 'G': grower = ON; break;
     case 'C': port = ON;
       serv= ON;
       equiv = ON;
       file_perms = ON;
       forward = ON;
       xwind = ON;
       rhoster_f=ON;
       rhoster= ON;
       ngroup= ON;
       complete= ON;
       majdomo = ON;
       grower = ON; break;
    case '-': break;
    case ' ': break;
    case 'p': break;
    case 's': break;
    case 'e': break;
    case 'b': break;
    case 'f': break;
    case 'x': break;
    case 'r': break;
    case 'v': break;
    case 'n': break;
    case 'm': break;
    case 'g': break;
    case 'c': break;
    default : printf("Unknown option: %c\n", commandstring[i]); exit(1);
      }
i++;
}


system("date > /tmp/.current_date");
input_fd2 = open ("/tmp/.current_date", O_RDONLY, 0);
bzero (input, sizeof(input));
i = 0;
j = 0;
k = 0;

while (read (input_fd2, buffer, 1) != 0)
  {
  strcat (input, buffer);
  j++;
  }

bzero (tempstring, sizeof (tempstring));
bzero (input2, sizeof (input2));

while (i < j)
  {
  if (input[i] == ' ') break;
  i++;
  }

i++;

tempstring[0] = input[i];
i++;
tempstring[1] = input[i];
i++;
tempstring[2] = input[i];
i++;


while (i < j)
  {
  if (input[i] != ' ') break;
  i++;
  }

tempstring[3] = '-';

k = 4;
```

```
  while (input[i] != ' ')
    {
    tempstring[k] = input[i];
    k++;
    i++;
    }

  i++;
  l = 0;

  while (input[i] != ' ')
    {
    input2[l] = input[i];
    i++;
    l++;
    }


  i++;

  tempstring[k] = '-';
  k++;

  while (input[i] != ' ') i++;

  i++;

  while (input[i] != '\n')
    {
    tempstring[k] = input[i];
    k++;
    i++;
    }


  tempstring[k] = '-';

  strcat (tempstring, input2);

/* printf("tempstring = %s\n", tempstring, input2);  */
  sprintf (input3, "./proj %s > sands.%s", commandstring, tempstring);
  system(input3);
  unlink(".current_date");

  }
```

test_exrc.c


```
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
if (forward == ON) {
       printf("Testing for .exrc files....");
       fflush(stdout);


       chdir ("/");

       if (grower == ON)
         {
          system ("/bin/find / -name '.exrc' > /tmp/garbage 2>&1");
          system ("grep '.exrc' /tmp/garbage > /tmp/doug_exrc");
          unlink ("/tmp/garbage");
          }
       else
         {
         system ("/bin/find . \\( \\! -fstype nfs -o -prune \\) -name '.exrc' > /tmp/dou
g_exrc");
         }

       input_fd = open ("/tmp/doug_exrc", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0)
           {
           strcat(input, buffer);
           };

       close (input_fd);

       unlink ("/tmp/doug_exrc");

 /*****************************************************/

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "\0") == 0) {
          printf("DONE.\n");
          printf("-------------------------------------------------------------\n");
          }

       else

          {
           printf("\n");
           printf("SANDS ALERT >> The following .exrc files were found.\n");
           printf("               Consider using the EXINIT environment \n");
           printf("               variable to disable .exrc functionality.\n");
           printf("\nHere are the absolute path names that were found: \n");
           printf("\n%s\n\n", input);

           if (rhoster_f == ON) {


           /*****************************************************/
           /*****************************************************/


           bzero(buffer, sizeof(buffer));

           i = 0;
           k = 0;

           while (i < strlen(input)) {
             sprintf(tempstring, "%c", input[i]);
             strcat(buffer, tempstring);


             if (strcmp(tempstring, "\n") == 0)
               {

               buffer[k] = '\0';
               buffer[k-5] = '\0';

               sprintf(tempstring, "ls -al %s | grep '.exrc' > /tmp/doug_exrc_security",
buffer);

               buffer[k-5] = '.';

               system (tempstring);
               input_fd = open ("/tmp/doug_exrc_security", O_RDONLY, 0);

               bzero(input3, sizeof(input3));
               bzero(input2, sizeof(input2));

               while (read (input_fd, input3, 1) != 0)
                 {
                 strcat(input2, input3);
                 };

               close (input_fd);
               unlink ("/tmp/doug_exrc_security");

               input2[10] = '\0';

               if (strcmp(input2, "-rw-------") != 0)
                 {
                 printf("\n");
                 printf("SANDS ALERT >> %s\n", buffer);
                 printf("               does not have permissions 600.\n");
                 printf("               It has permissions %s\n", input2);
                 }

               /*****************************************************/


               j = 0;

               while (j < 8)
                 {
                 tempstring[j] = input2[j+15];
                 j++;
                 }

               tempstring[j] = '\0';

               printf("\n");
               printf("SANDS NOTE  >> %s\n", buffer);
               printf("               has OWNER %s\n", tempstring);
               bzero(buffer, sizeof(buffer));
               k=-1;
               }

             i++;
```

```
            k++;
          }

        }

        printf("DONE.\n");
        printf("-------------------------------------------------------------\n");

    }

}
```

```
/*************************************************/
/*************************************************/
/*************************************************/
/*************************************************/

      printf("Testing File Permissions....");
      fflush(stdout);

      chdir("/etc");
      strcpy (tempstring, "passwd\0");
      system ("ls -l | grep passwd > /tmp/doug_passwd");
      input_fd = open ("/tmp/doug_passwd", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*************************************************/
/* This is to determine if passwd is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep passwd > /tmp/doug_passwd", tempstring);

         /* printf("This is the command := %s\n", input); */

         system (input);

         }
/*************************************************/

      input_fd = open ("/tmp/doug_passwd", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/passwd does not have permissions 644.\n");
          printf("               passwd has permissions %s\n", input);
      }


/*************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/passwd does not have owner ROOT.\n");
          printf("               passwd has OWNER %s\n", tempstring);
          }



/*************************************************/
/*************************************************/
/*************************************************/

      strcpy (tempstring, "utmp\0");
      system ("ls -l | grep utmp > /tmp/doug_utmp");
      input_fd = open ("/tmp/doug_utmp", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*************************************************/
/* This is to determine if utmp is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep utmp > /tmp/doug_utmp", tempstring);
```

```
        /* printf("This is the command := %s\n", input); */

        system (input);

        }
/*************************************************/

      input_fd = open ("/tmp/doug_utmp", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
          };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/utmp does not have permissions 644.\n");
          printf("               utmp has permissions %s\n", input);
      }


/*************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/utmp does not have owner ROOT.\n");
          printf("               utmp has OWNER %s\n", tempstring);
          }


/*************************************************/
/*************************************************/
/*************************************************/


      strcpy (tempstring, "motd\0");
      system ("ls -l | grep motd > /tmp/doug_motd");
      input_fd = open ("/tmp/doug_motd", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
          };
      close (input_fd);

/*************************************************/
/* This is to determine if motd is a link! */
      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep motd > /tmp/doug_motd", tempstring);

         /* printf("This is the command := %s\n", input); */

         system (input);

         }
/*************************************************/

      input_fd = open ("/tmp/doug_motd", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
          };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/motd does not have permissions 644.\n");
          printf("               motd has permissions %s\n", input);
      }


/*************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/motd does not have owner ROOT.\n");
          printf("               motd has OWNER %s\n", tempstring);
          }
```

```
/*************************************************/
/*************************************************/
/*************************************************/

      strcpy (tempstring, "syslog.pid\0");
      system ("ls -l | grep syslog.pid > /tmp/doug_syslog.pid");
      input_fd = open ("/tmp/doug_syslog.pid", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*************************************************/
/* This is to determine if syslog.pid is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep syslog.pid > /tmp/doug_syslog.pid", tempstring
);

         /* printf("This is the command :- %s\n", input); */

         system (input);

         }
/*************************************************/

      input_fd = open ("/tmp/doug_syslog.pid", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/syslog.pid does not have permissions 644.\n");
          printf("                syslog.pid has permissions %s\n", input);
      }


/*************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root     ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/syslog.pid does not have owner ROOT.\n");
          printf("                syslog.pid has OWNER %s\n", tempstring);
          }


/*************************************************/
/*************************************************/
/*************************************************/
/*************************************************/
/*************************************************/
/*************************************************/

      j = 1;

      while (j <= 7)
         {
         bzero(tempstring, sizeof(tempstring));
         if (j == 1) strcpy (tempstring, "/etc\0");
         if (j == 2) strcpy (tempstring, "/bin\0");
         if (j == 3) strcpy (tempstring, "/usr/bin\0");
         if (j == 4) strcpy (tempstring, "/sbin\0");
         if (j == 5) strcpy (tempstring, "/usr/sbin\0");
         if (j == 6) strcpy (tempstring, "/tmp\0");
         if (j == 7) strcpy (tempstring, "/var/tmp\0");

         sprintf(buffer2, "ls -ld %s > /tmp/doug_files", tempstring);
         system (buffer2);
         sprintf(buffer2, "/tmp/doug_files");

         input_fd = open (buffer2, O_RDONLY, 0);

         bzero(input, sizeof(input));
         bzero(buffer, sizeof(buffer));

         while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
         };
         close (input_fd);

         if (j > 5)
           {
           input[10] = '\0';

           if (strcmp(input, "drwxrwxrwt") != 0)
             {
             printf("\n");
             printf("SANDS ALERT >> %s does not have permissions 1777.\n",tempstring);
```

```
            printf("                %s has permissions %s\n",tempstring, input);
            }
          }


        i = 0;

        while (i < 8)
          {
          input3[i] = input[i+15];
          i++;
          }

        input3[i] = '\0';

        if (strcmp(input3, "root    ") != 0)
          {
          printf("\n");
          printf("SANDS ALERT >> %s does not have owner ROOT.\n", tempstring);
          printf("                %s has OWNER %s\n", tempstring, input3);
          }

        unlink("/tmp/doug_files");
        j++;
}

 /*****************************************************/
 /*****************************************************/
 /*****************************************************/


        printf("\n-------------------------------------------------------------\n");
        printf("SANDS found the following suspicious files in /dev:\n");
        printf("\n");

        fflush(stdout);

        system("/bin/find /dev -type f -exec ls -l {} \\;");
        printf("\n");

        if (file_perms == ON)
           {
           printf("\n-------------------------------------------------------------\n");
           printf("SANDS found the following unexpected files outside of /dev:\n");
           printf("\n");
           fflush(stdout);

           if (grower == ON)
             {
             system ("/bin/find / \\( -type b -o -type c \\) -print | grep -v '^/dev/'
");
             }
           else
             {
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) \\( -type b -o -ty
pe c \\) -print | grep -v '^/dev/'");
             }
           }


        if (file_perms == ON)
           {
           printf("\n-------------------------------------------------------------\n");
           printf("SANDS found the following world-writeable files:\n");
           printf("\n");
           fflush(stdout);

           if (grower == ON)
             {
             system ("/bin/find / -type f \\( -perm -2 -o -perm 20 \\) ");
             system ("/bin/find / -type d \\( -perm -2 -o -perm 20 \\) ");
             }
           else
             {
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type f \\( -perm -
2 -o -perm 20 \\) ");
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type d \\( -perm -
2 -o -perm 20 \\) ");
             }

           printf("\n-------------------------------------------------------------\n");
           printf("SANDS found the following SUID and SGID files:\n");
           printf("\n");
           fflush(stdout);

           if (grower == ON)
             {
             system ("/bin/find / -type f \\( -perm -004000 -o -perm 002000 \\)");
             system ("/bin/find / -type d \\( -perm -004000 -o -perm 002000 \\)");
             }
           else
             {
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type f \\( -perm -
004000 -o -perm 002000 \\) ");
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type d \\( -perm -
004000 -o -perm 002000 \\) ");
             }

           }


 /*****************************************************/
 /*****************************************************/
 /*****************************************************/

        if (file_perms == ON)
           {
           printf("\n-------------------------------------------------------------\n");
           printf("SANDS recommends changing the ownership of the following files\n");
           printf("from bin to root:\n");
           printf("\n");
           fflush(stdout);

           if (grower == ON)
             {
             system ("/bin/find / -type f \\( -perm -4 -a ! -perm -2 \\) -user bin");
             system ("/bin/find / -type f \\( -perm -4 -a ! -perm -20 \\) -user bin");
             }
           else
             {
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type f \\( -perm -
4 -a ! -perm -2 \\) -user bin");
             system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type f \\( -perm -
4 -a ! -perm -20 \\) -user bin");
             }
           }

 /*****************************************************/
```

```
/*****************************************************/
/*****************************************************/

        unlink("/tmp/doug_utmp");
        unlink("/tmp/doug_passwd");
        unlink("/tmp/doug_motd");
        unlink("/tmp/doug_syslog.pid");
        printf("DONE.\n");
        printf("------------------------------------------------------------\n");
```

test_forward.c


```
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
if (forward == ON) {

       printf("Testing for .forward files....");
       fflush(stdout);


       chdir ("/");

       if (grower == ON)
         {
          system ("/bin/find / -name '.forward' > /tmp/garbage 2>&1");
          system ("grep '.forward' /tmp/garbage > /tmp/doug_forward");
          unlink ("/tmp/garbage");
          }
       else
         {
         system ("/bin/find . \\( \\! -fstype nfs -o -prune \\) -name '.forward' > /tmp/
doug_forward");
         }

       input_fd = open ("/tmp/doug_forward", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0)
           {
           strcat(input, buffer);
           };

       close (input_fd);

       unlink ("/tmp/doug_forward");

 /*****************************************************/

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "\0") == 0) {
          printf("DONE.\n");
          printf("------------------------------------------------------------\n");
          }

       else

          {
           printf("\n");
           printf("SANDS ALERT >> The following .forward files were found.\n");
           printf("               Ensure that they do not allow a normal\n");
           printf("               user priveleged access.  Please refer to\n");
           printf("               AUSCERT SA-93.10\n");
           printf("\nHere are the absolute path names that were found: \n");
           printf("\n%s\n\n", input);

           if (rhoster_f == ON) {


           /*****************************************************/
           /*****************************************************/

           bzero(buffer, sizeof(buffer));

           i = 0;
           k = 0;
           l = 0;


           while (i < strlen(input)) {
             sprintf(tempstring, "%c", input[i]);
             strcat(buffer, tempstring);


             if (strcmp(tempstring, "\n") == 0)
               {

               buffer[k] = '\0';
               buffer[k-8] = '\0';

               sprintf(tempstring, "ls -al %s | grep '.forward' > /tmp/doug_forward_secur
ity", buffer);

               buffer[k-8] = '.';

               system (tempstring);
               input_fd = open ("/tmp/doug_forward_security", O_RDONLY, 0);

               bzero(input3, sizeof(input3));
               bzero(input2, sizeof(input2));

               while (read (input_fd, input3, 1) != 0)
                 {
                 strcat(input2, input3);
                 };

               close (input_fd);
               unlink ("/tmp/doug_forward_security");

               input2[10] = '\0';

               if (strcmp(input2, "-rw-------") != 0)
                 {
                 printf("\n");
                 printf("SANDS ALERT >> %s\n", buffer);
                 printf("               does not have permissions 600.\n");
                 printf("               It has permissions %s\n", input2);
                 }

               /*****************************************************/


               j = 0;

               while (j < 8)
                 {
                 tempstring[j] = input2[j+15];
                 j++;
                 }

               tempstring[j] = '\0';

               printf("\n");
               printf("SANDS NOTE  >> %s\n", buffer);
               printf("               has OWNER %s\n", tempstring);
               bzero(buffer, sizeof(buffer));
```

```
                k--1;
                }

              i++;
              k++;

          }

        }

        printf("DONE.\n");
        printf("------------------------------------------------------------\n");

    }

}
```

```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

      printf("Testing /etc/hosts.equiv....");

      bzero(tempstring, sizeof(tempstring));

      strcpy (tempstring, "hosts.equiv\0");

      system ("ls -l | grep hosts.equiv > /tmp/doug_hosts.equiv");
      /* system ("more hosts.equiv > /tmp/doug_more_hosts.equiv"); */

      input_fd = open ("/tmp/doug_hosts.equiv", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*****************************************************/

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "\0") == 0) {
         printf("DONE.\n");
         printf("------------------------------------------------------------\n");
         }

      else

         {
          printf("\n");
          printf("SANDS NOTE  >> hosts.equiv was found.  If it is not \n");
          printf("               needed, recommend that it be removed.\n");
          printf("               If needed, ensure that the hosts are in\n");
          printf("               your domain and under your management.\n");
          printf("               Also, use fully qualified hostnames.\n");

          system ("more hosts.equiv > /tmp/doug_more_hosts.equiv");

/*****************************************************/
/* This is to determine if hosts.equiv is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {
         /* printf("We have a link!\n"); */

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep hosts.equiv > /tmp/doug_hosts.equiv", tempstrin
g);
         sprintf(input2, "more %s > /tmp/doug_more_hosts.equiv", tempstring);


         /* printf("This is the command := %s\n", input);  */

         system (input);
         system (input2);

         }


/*****************************************************/

      input_fd = open ("/tmp/doug_hosts.equiv", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-------") != 0) {
          printf("\n");
          printf("SANDS ALERT >> hosts.equiv does not have permissions 600.\n");
          printf("               hosts.equiv has permissions %s\n", input);
      }


/*****************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root     ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> hosts.equiv does not have owner ROOT.\n");
          printf("               hosts.equiv has OWNER %s\n", tempstring);
          }

/*****************************************************/
/*****************************************************/

      input_fd2 = open ("/tmp/doug_more_hosts.equiv", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      plus_flag = 0;
```

```
i = 0;
j = 0;

k = 0;
l = 1;

while (read (input_fd2, buffer, 1) != 0) {
      strcat(input, buffer);

      sprintf(tempstring, "%c", buffer[0]);

      if (strcmp(tempstring, "\n") == 0)
         {
         l = l + 1;
         k = 0;
         }
      else
        {
        k = k + 1;
        if (strcmp(tempstring, " ") != 0) plus_flag = 0;
        }

      if ((strcmp(tempstring, "\n") == 0) && (plus_flag == 1)) {
         printf("\n");
         printf("SANDS ALERT >> There is a '+' in hosts.equiv at\n");
         printf("               ROW = %d and COLUMN = %d \n", l-1, i);
         printf("               DO NOT have a '+' by itself in this file! \n");
         plus_flag = 0;
         }

      if ((strcmp(tempstring, " ") == 0) && (plus_flag == 1)) {
         printf("\n");
         printf("SANDS ALERT >> There is a '+' in hosts.equiv at\n");
         printf("               ROW = %d and COLUMN = %d \n", l, i);
         printf("               DO NOT have a '+' by itself in this file! \n");
         plus_flag = 0;
         }

      if (strcmp(tempstring, "+") == 0) {
          plus_flag = 1;
          i = k;
          }

      if (strcmp(tempstring, "!") == 0) {
         printf("\n");
         printf("SANDS ALERT >> There is a '!' in hosts.equiv at\n");
         printf("               ROW = %d and COLUMN = %d \n", l, k);
         printf("               There are no comments in /etc/hosts.equiv. \n");
         }


      if (strcmp(tempstring, "#") == 0) {
         printf("\n");
         printf("SANDS ALERT >> There is a '#' in hosts.equiv at\n");
         printf("               ROW = %d and COLUMN = %d \n", l, k);
         printf("               There are no comments in /etc/hosts.equiv. \n");
         }

     };


sprintf(tempstring, "%c", input[0]);

if (strcmp(tempstring, "-") == 0) {
         printf("\n");
         printf("SANDS ALERT >> The first character in hosts.equiv is '-' \n");
         printf("               Please refer to AUSCERT Advisory \n");
         printf("               CA-91:12.\n");
         }

/*************************************************/

      if (l > 10)  {
          printf("SANDS ALERT >> You have more than 10 entries in your hosts.equiv \n");
          printf("               file.  Recommend reducing the number of entries. \n");
          }

      close(input_fd2);
      unlink("/tmp/doug_hosts.equiv");
      unlink("/tmp/doug_more_hosts.equiv");
      printf("DONE.\n");
      printf("------------------------------------------------------------\n");


      }
```

```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

      if (equiv == ON) {
         printf("\nHere are the entries in the host.equiv file: \n");
         printf("\n%s",input);
         printf("\nDONE.\n");
         printf("------------------------------------------------------------\n");

         }
      else
         {
         printf("\n");
         printf("------------------------------------------------------------\n");
         }
      bzero(tempstring, sizeof(tempstring));
```

test_hosts.lpd.c


```
/*****************************************************/
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/

       printf("Testing /etc/hosts.lpd....");

       bzero(tempstring, sizeof(tempstring));

       chdir ("/etc/");

       strcpy (tempstring, "hosts.lpd\0");

       system ("ls -l | grep hosts.lpd > /tmp/doug_hosts.lpd");
       /* system ("more hosts.lpd > /tmp/doug_more_hosts.lpd"); */

       input_fd = open ("/tmp/doug_hosts.lpd", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
            };
       close (input_fd);

 /*****************************************************/

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "\0") == 0) {
          printf("DONE.\n");
          printf("------------------------------------------------------------\n");
          }

       else

          {
           system ("more hosts.lpd > /tmp/doug_more_hosts.lpd");
 /*****************************************************/
 /* This is to determine if hosts.lpd is a link! */

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "l") == 0) {
          /* printf("We have a link!\n"); */

          bzero(tempstring, sizeof(tempstring));

          strncat(tempstring, strchr(input, '>'), 20);

          i = 0;

          while (i < 97) {
          tempstring[i] = tempstring[i+2];
             if (tempstring[i] == '\n') {
                tempstring[i] = '\0';
                break;
                }
             i++;
            }


          sprintf(input, "ls -l %s | grep hosts.lpd > /tmp/doug_hosts.lpd", tempstring);

          sprintf(input2, "more %s > /tmp/doug_more_hosts.lpd", tempstring);

          /* printf("This is the command := %s\n", input);  */

          system (input);
          system (input2);

          }


 /*****************************************************/

       input_fd = open ("/tmp/doug_hosts.lpd", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
            };

       input[10] = '\0';

       if (strcmp(input, "-rw-------") != 0) {
           printf("\n");
           printf("SANDS ALERT >> hosts.lpd does not have permissions 600.\n");
           printf("               hosts.lpd has permissions %s\n", input);
       }


 /*****************************************************/


       i = 0;

       while (i < 8) {
          tempstring[i] = input[i+15];
          i++;
         }

       tempstring[i] = '\0';

       if (strcmp(tempstring, "root     ") != 0) {
           printf("\n");
           printf("SANDS ALERT >> hosts.lpd does not have owner ROOT.\n");
           printf("               hosts.lpd has OWNER %s\n", tempstring);
           }


 /*****************************************************/
 /*****************************************************/

       input_fd2 = open ("/tmp/doug_more_hosts.lpd", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));


       i = 0;
       j = 0;

       k = 0;
       l = 1;
```

```
      while (read (input_fd2, buffer, 1) != 0) {
            strcat(input, buffer);

            sprintf(tempstring, "%c", buffer[0]);

            if (strcmp(tempstring, "\n") == 0)
               {
               l = l + 1;
               k = 0;
               }
            else
              {
              k = k + 1;
              }


            if (strcmp(tempstring, "!") == 0) {
               printf("\n");
               printf("SANDS ALERT >> There is a '!' in hosts.lpd at\n");
               printf("               ROW = %d and COLUMN = %d \n", l, k);
               printf("               There are no comments in /etc/hosts.lpd. \n");
               }


            if (strcmp(tempstring, "#") == 0) {
               printf("\n");
               printf("SANDS ALERT >> There is a '#' in hosts.lpd at\n");
               printf("               ROW = %d and COLUMN = %d \n", l, k);
               printf("               There are no comments in /etc/hosts.lpd. \n");
               }

           };


      sprintf(tempstring, "%c", input[0]);

      if (strcmp(tempstring, "-") == 0) {
          printf("\n");
          printf("SANDS ALERT >> The first character in hosts.lpd is '-' \n");
          printf("               Please refer to AUSCERT Advisory \n");
          printf("               CA-93:05.\n");
          }

/*****************************************************/

      close(input_fd2);
      unlink("/tmp/doug_more_hosts.lpd");
      printf("DONE.\n");
      printf("------------------------------------------------------------\n");

          }

      unlink("/tmp/doug_hosts.lpd");
```

test_inetd.conf.c

```
      printf("Testing /etc/inetd.conf....");

      chdir("/etc");
      strcpy (tempstring, "inetd.conf\0");
      system ("ls -l | grep inetd.conf > /tmp/doug_inetd.conf");
      input_fd = open ("/tmp/doug_inetd.conf", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*****************************************************/
/* This is to determine if inetd.conf is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep inetd.conf > /tmp/doug_inetd.conf", tempstring
);
         sprintf(input2, "more %s | grep tftp > /tmp/doug_tftp", tempstring);
         sprintf(buffer2, "more %s | grep rexd/ > /tmp/doug_rexd", tempstring);
         sprintf(input3, "more %s > /tmp/doug_inetd_services", tempstring);

         /* printf("This is the command := %s\n", input); */

         system (input);

         }
/*****************************************************/

      input_fd = open ("/tmp/doug_inetd.conf", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-------") != 0) {
          printf("\n");
          printf("SANDS ALERT >> inetd.conf does not have permissions 600.\n");
          printf("              inetd.conf has permissions %s\n", input);
      }


/*****************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> inetd.conf does not have owner ROOT.\n");
          printf("              inetd.conf has OWNER %s\n", tempstring);
          }


/*****************************************************/

      system(input2);

      input_fd2 = open ("/tmp/doug_tftp", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd2, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "#") != 0) {
          printf("\n");
          printf("SANDS ALERT >> tftp in inetd.conf is not commented out.\n");
          printf("              Recommend commenting it out.\n");
          printf("              If tftp is required, please read AUSCERT\n");
          printf("              Advisory CA-93:05.\n");
          }

/*****************************************************/
/*****************************************************/

      system(buffer2);

      input_fd2 = open ("/tmp/doug_rexd", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd2, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "#") != 0) {
          printf("\n");
          printf("SANDS ALERT >> rexd in inetd.conf is not commented out.\n");
          printf("              Recommend commenting it out.  Rexd servers\n");
```

```
        printf("              have little or no security in their design.\n");
        printf("              Intruders can use this service to execute\n");
        printf("              commands as any user. \n");
        }

/*************************************************/


      close(input_fd2);
      unlink("/tmp/doug_inetd.conf");
      unlink("/tmp/doug_tftp");
      unlink("/tmp/doug_rexd");
      printf("DONE.\n");
      printf("-------------------------------------------------------------\n");
```

test_inetd.conf.verbose.c

```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

      if (serv == ON) {
          system (input3);
          input_fd2 = open ("/tmp/doug_inetd_services", O_RDONLY, 0);

          bzero(input, sizeof(input));
          bzero(buffer, sizeof(buffer));

          l = 0;
          i = 1;
          k = 0;
          j = 0;

          while (read (input_fd2, buffer, 1) != 0) {
             sprintf(tempstring, "%c", buffer[0]);

             if (strcmp(tempstring, "\n") == 0)
               {
               k = 0;
               l = 0;
               }

             if ((strcmp(tempstring, "#") == 0) && (k == 0))
               {
               k = 1;
               l = 0;
               }

             if ((strcmp(tempstring, "#") != 0) &&
                 (strcmp(tempstring, "\n") != 0) &&
                 (k == 0))
               {
               l = 1;
               }

             if (((strcmp(tempstring, " ") == 0)  ||
                 (strcmp(tempstring, "\t") == 0)) &&
                 (l == 1))
                 {
                  if (i == 4) {
                        strcat(input, "\n");
                        i = 1;
                        j = 0;
                        }
                  else {
                        if (j > 7) strcat (input, "\t");
                        else strcat (input, "\t\t");
                        i = i + 1;
                        j = 0;
                        }

                  l = 0;
                  k = 1;
                 }

             if (l == 1) {
                 strcat(input, tempstring);
                 j = j + 1;
                 }
                }

/*****************************************************/

      close(input_fd2);
      unlink("/tmp/doug_inetd_services");

      printf("Here are the services in the inetd.conf file: \n");
      printf("\n%s\n\n",input);
      printf("DONE.\n");
      printf("-------------------------------------------------------------\n");
      bzero(tempstring, sizeof(tempstring));

          }
      else
          {
          printf("\n");
          printf("-------------------------------------------------------------\n");
          }
```

test_majordomo_receive.c

```
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
if (majdomo == ON) {

       printf("Testing majordomo version....");

system ("echo $LOGNAME > /tmp/nammerr");

input_fd = open ("/tmp/nammerr", O_RDONLY, 0);

bzero(input, sizeof(input));
bzero(buffer, sizeof(buffer));
bzero(tempstring, sizeof(tempstring));

while (read (input_fd, buffer, 1) != 0)
{
    sprintf(tempstring, "%c", buffer[0]);
    if (strcmp(tempstring, "\n") == 0)
      {
      strcat(input, "\0");
      }
    else strcat(input,buffer);
  }


close (input_fd);
unlink ("/tmp/nammerr");

sprintf(tempstring, "more /var/mail/%s | grep Chapman > /tmp/test_majordomo_version", in
put);
/* printf("%s\n", input);
printf("%s", tempstring);
fflush(stdout); */
system (tempstring);

input_fd2 = open ("/tmp/test_majordomo_version", O_RDONLY, 0);

bzero(input, sizeof(input));
bzero(input2, sizeof(input2));
bzero(buffer, sizeof(buffer));
bzero(tempstring, sizeof(tempstring));

i = 0;
j = 0;
plus_flag =0;


while (read (input_fd2, buffer, 1) != 0)
{
  strcat(input, buffer);
  if (plus_flag == 1)
    {
    strcat(input2, buffer);
    }
  sprintf(tempstring, "%c", buffer[0]);
  if (strcmp(tempstring, "v") == 0)
    {
    read (input_fd2, buffer, 1);
    sprintf(tempstring, "%c", buffer[0]);
    i++;
      if (strcmp(tempstring, "e") == 0)
        {
        read (input_fd2, buffer, 1);
        sprintf(tempstring, "%c", buffer[0]);
        i++;
          if (strcmp(tempstring, "r") == 0)
            {
            read (input_fd2, buffer, 1);
            sprintf(tempstring, "%c", buffer[0]);
            i++;
              if (strcmp(tempstring, "s") == 0)
                {
                read (input_fd2, buffer, 1);
                sprintf(tempstring, "%c", buffer[0]);
                i++;
                  if (strcmp(tempstring, "i") == 0)
                    {
                    read (input_fd2, buffer, 1);
                    sprintf(tempstring, "%c", buffer[0]);
                    i++;
                      if (strcmp(tempstring, "o") == 0)
                        {
                        read (input_fd2, buffer, 1);
                        sprintf(tempstring, "%c", buffer[0]);
                        i++;
                          if (strcmp(tempstring, "n") == 0)
                            {
                            plus_flag = 1;

                            }
                        }
                    }
                }
            }
        }
    }
  i++;
  }

unlink("/tmp/test_majordomo_version");
fflush(stdout);

if (strcmp(input, "\0") == 0)
{
  printf("\n");
  printf("SANDS encountered an error checking the version of majordomo.\n");
  printf("Majordomo may not be runnning, or SANDS may have mishandled\n");
  printf("your mailer.  To check the version manually, send mail to \n");
  printf("'majordomo' with the message text of 'help' and it will mail\n");
  printf("the version back to you.\n");
  printf("DONE.\n");
  printf("------------------------------------------------------------\n");
  }

else
{
  printf("\n");
  printf("Your majordomo version is %s\n",input2);

  i = 0;
  input2[5] = '\0';
  while (i < 6)
    {
    input2[i] = input2[i+1];
    i++;
    }
```

```
  if (atof(input2) > 1.91)
    {
    printf("DONE.\n");
    printf("-------------------------------------------------------------\n");
    }

  else
    {
    printf("\n");
    printf("SANDS ALERT >> Your majordomo version is less than \n");
    printf("               1.91.  This needs to be updated.\n");
    printf("               Please see AUSCERT Advisory SA-94.03\n");
    printf("DONE.\n");
    printf("-------------------------------------------------------------\n");

    }
}
}
```

test_majordomo_send.c


```
if (majdomo == ON)
{
  chdir ("/tmp/");

  system ("echo help > helpperr");

  system ("mail majordomo < helpperr");

  unlink ("helpperr");
  }
```

test_netgroup.c


```
/*****************************************************/
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/

       printf("Testing /etc/netgroup....");

       bzero(tempstring, sizeof(tempstring));

       chdir ("/etc/");

       strcpy (tempstring, "netgroup\0");

       system ("ls -l | grep netgroup > /tmp/doug_netgroup");
       /* system ("more hosts.lpd > /tmp/doug_more_hosts.lpd"); */

       input_fd = open ("/tmp/doug_netgroup", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
            };
       close (input_fd);

       unlink("/tmp/doug_netgroup");

/*****************************************************/

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "\0") == 0) {
          printf("DONE.\n");
          printf("------------------------------------------------------------\n");
          }

       else

          {
          printf("\n");
          printf("SANDS NOTE  >> netgroup was found.  SANDS assumes that \n");
          printf("               you are running NIS or NIS+.  This is\n");
          printf("               a reminder that each netgroup should \n");
          printf("               contain only hostnames or only usernames.\n");


          system ("more netgroup > /tmp/doug_more_netgroup");

/*****************************************************/
/* This is to determine if netgroup is a link! */

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "l") == 0) {
          /* printf("We have a link!\n"); */

          bzero(tempstring, sizeof(tempstring));

          strncat(tempstring, strchr(input, '>'), 20);

          i = 0;

          while (i < 97) {
          tempstring[i] = tempstring[i+2];
             if (tempstring[i] == '\n') {
                tempstring[i] = '\0';
                break;
                }
             i++;
            }


          sprintf(input, "ls -l %s | grep netgroup > /tmp/doug_netgroup", tempstring);
          sprintf(input2, "more %s > /tmp/doug_more_netgroup", tempstring);


          /* printf("This is the command := %s\n", input);  */

          system (input);
          system (input2);

          }


/*****************************************************/

       input_fd = open ("/tmp/doug_netgroup", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
            };

       input[10] = '\0';

       if (strcmp(input, "-rw-------") != 0) {
           printf("\n");
           printf("SANDS ALERT >> netgroup does not have permissions 600.\n");
           printf("               netgroup has permissions %s\n", input);
       }


/*****************************************************/


       i = 0;

       while (i < 8) {
          tempstring[i] = input[i+15];
          i++;
         }

       tempstring[i] = '\0';

       if (strcmp(tempstring, "root     ") != 0) {
           printf("\n");
           printf("SANDS ALERT >> netgroup does not have owner ROOT.\n");
           printf("               netgroup has OWNER %s\n", tempstring);
           }

          bzero(input, sizeof(input));
          bzero(buffer, sizeof(buffer));

          while (read (input_fd2, buffer, 1) != 0)
            {
            strcat(input, buffer);
            }
```

```
close(input_fd2);

unlink("/tmp/doug_more_netgroup");
printf("DONE.\n");
printf("------------------------------------------------------------\n");


}
```

test_netgroup.verbose.c


```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

      if (ngroup == ON) {
         printf("\nHere are the entries in the netgroups file: \n");
         printf("\n%s",input);
         printf("\nDONE.\n");
         printf("-------------------------------------------------------------\n");

         }
      else
         {
         printf("\n");
         printf("-------------------------------------------------------------\n");
         }
      bzero(tempstring, sizeof(tempstring));
```

```
/*************************************************/
/*************************************************/
/*************************************************/
/*************************************************/

      printf("Testing NFS....");

      strcpy (tempstring, "dfstab\0");
      system ("ls -l /etc/dfs | grep dfstab > /tmp/doug_dfstab");
      input_fd = open ("/tmp/doug_dfstab", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*************************************************/
/* This is to determine if dfstab is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep dfstab > /tmp/doug_dfstab", tempstring);

         /* printf("This is the command :- %s\n", input); */

         system (input);

         }
/*************************************************/

      input_fd = open ("/tmp/doug_dfstab", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/dfs/dfstab does not have permissions 644.\n");
          printf("                dfstab has permissions %s\n", input);
          }


/*************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> /etc/dfs/dfstab does not have owner ROOT.\n");
          printf("                dfstab has OWNER %s\n", tempstring);
          }


      if (rhoster_f)
          {
          printf("\nThe following systems are in the ");
          fflush(stdout);
          system("showmount -e");
          }

/*************************************************/
/*************************************************/
/*************************************************/
      system ("more /etc/dfs/dfstab | grep -v '#'> /tmp/doug_dfs");
      input_fd = open ("/tmp/doug_dfs", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

      system ("hostname > /tmp/doug_hostname");
      input_fd = open ("/tmp/doug_hostname", O_RDONLY, 0);

      bzero(input2, sizeof(input2));
      bzero(input3, sizeof(input3));

      i = 0;

      while (read (input_fd, input3, 1) != 0) {
            strcat(input2, input3);
            i++;
           };
      close (input_fd);

      i--;
      input2[i] = '\0';

       i = 0;
       j = 0;
       k = 0;
       l = OFF;
```

```
while (i < strlen(input))
    {
    if (input[i] == '=')
      {
      j = i+1;

      while (input[i] != '.')
        {
        if ((input[i] == ' ') || (input[i] == ',') ||
            (input[i] == '\t') || (input[i] == '\n'))
          {
          k = 0;
          bzero(tempstring, sizeof(tempstring));

          while (j < i)
            {
            tempstring[k] = input[j];
            k++;
            j++;
            }

          if (strcmp(tempstring, input2) == 0) 1 = ON;

          printf("\n");
          printf("SANDS ALERT >> USE fully qualified hostnames in \n");
          printf("               /etc/dfs/dfstab. The host %s\n", tempstring);
          printf("               is not fully qualified.\n");
          break;
          }
        i++;
        }
      }
    i++;
    }

    if (1 == ON)
       {
          printf("\n");
          printf("SANDS ALERT >> /etc/dfs/dfstab contains the localhost\n");
          printf("               %s.  This should not be allowed.\n", input2);
          }


unlink("/tmp/doug_dfstab");
unlink("/tmp/doug_dfs");
unlink("/tmp/doug_hostname");
printf("DONE.\n");
printf("-------------------------------------------------------------\n");
```

```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

      if (port == ON) {

          j = system ("ls > /dev/null");
          i = system ("rpcinfo -s > /tmp/test_doug_rpcinfo");

          if (i != j) {
             i = system ("/usr/bin/rpcinfo -s > /tmp/test_doug_rpcinfo");
             }

          if (i != j) {
             i = system ("/usr/sbin/rpcinfo -s > /tmp/test_doug_rpcinfo");
             }

          if (i != j)
            {
            printf("\nSANDS is unable to access the necessary commands\n");
            printf("to determine what services are registered.\n");
            printf("Please check your configuration.\n");
            exit(0);
            }

          input_fd2 = open ("/tmp/test_doug_rpcinfo", O_RDONLY, 0);

          bzero(input, sizeof(input));
          bzero(buffer, sizeof(buffer));

          l = 0;
          i = 0;
          k = 0;
          j = 100;
          plus_flag = 0;

          if (rhoster_f == ON)
            {
            bzero (input2, sizeof(input2));
            while (read (input_fd2, buffer, 1) != 0) {
                  strcat (input2, buffer);
                 }
            }

          while (read (input_fd2, buffer, 1) != 0) {
             sprintf(tempstring, "%c", buffer[0]);

             if (l == 0) k = k + 1;
             i = i + 1;

             if ((i > j + 2) && (plus_flag == 1))
               {
               strcat(input, tempstring);
               }

             if ((i > j + 2) && (strcmp(tempstring, " ") == 0)
                 && (plus_flag == 1))
               {
               k = k + 1;
               if (i > j + 9)  strcat(input, "\t");
               else strcat(input, "\t\t");

               if (k > 3)
                 {
                 k = 0;
                 strcat(input, "\n");
                 }
               i = 0;

               }



          if ((strcmp(tempstring, "s") == 0) &&
              (l == 0))
            {
            j = k;
            read (input_fd2, buffer, 1);
            sprintf(tempstring, "%c", buffer[0]);
            if (strcmp(tempstring, "e") == 0)
              {
              read (input_fd2, buffer, 1);
              sprintf(tempstring, "%c", buffer[0]);
              if (strcmp(tempstring, "r") == 0)
                {
                read (input_fd2, buffer, 1);
                sprintf(tempstring, "%c", buffer[0]);
                if (strcmp(tempstring, "v") == 0)
                  {
                  read (input_fd2, buffer, 1);
                  sprintf(tempstring, "%c", buffer[0]);

                  if (strcmp(tempstring, "i") == 0)
                    {
                    read (input_fd2, buffer, 1);
                    sprintf(tempstring, "%c", buffer[0]);
                    if (strcmp(tempstring, "c") == 0)
                      {
                      read (input_fd2, buffer, 1);
                      sprintf(tempstring, "%c", buffer[0]);
                      if (strcmp(tempstring, "e") == 0)
                        {
                        l = 1;
                        printf("\n");
                        k = 0;
                        }
                      }
                    }
                  }
                }
              }
            }


          if (l != 1) j = 100;


          if ((strcmp(tempstring, "\n") == 0) &&
              (l == 0))
            {
            printf("SANDS has encountered unexpected format - TEST FAILED!!\n");
            exit(0);
            }

          if (strcmp(tempstring, "\n") == 0)
              {
              i = 0;
```

```
                plus_flag = 1;
                }

}

/*              if (((strcmp(tempstring, " ") == 0)  ||
                    (strcmp(tempstring, "\t") == 0)) &&
                    (l == 1))
                    {
                     if (i == 4) {
                           strcat(input, "\n");
                           i = 1;
                           j = 0;
                           }
                     else {
                           if (j > 7) strcat (input, "\t");
                           else strcat (input, "\t\t");
                           i = i + 1;
                           j = 0;
                           }

                     l = 0;
                     k = 1;
                    }

              if (l == 1) {
                    strcat(input, tempstring);
                    j = j + 1;
                    }

              }



 /*************************************************/

       close(input_fd2);
       unlink("/tmp/test_doug_rpcinfo");

       printf("Here are the services registered with the portmapper: \n");
       if (rhoster_f == OFF) printf("\n%s\n\n",input);
       else if (rhoster_f == ON) printf("\n%s\n\n",input2);
       printf("DONE.\n");
       printf("-------------------------------------------------------------\n");
       bzero(tempstring, sizeof(tempstring));

           }
       else
           {
           printf("\n");
           printf("-------------------------------------------------------------\n");
           }
```

```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/
if (rhoster == ON){
      printf("Testing for .rhosts files....");

      bzero(tempstring, sizeof(tempstring));

      strcpy (tempstring, ".rhosts\0");

      chdir("/");

      fflush(stdout);

      if (grower == ON)
        {
         system ("/bin/find . -name '.rhosts' > /tmp/garbage 2>&1");
         system ("grep '.rhosts' /tmp/garbage > /tmp/doug_rhosts");
         unlink ("/tmp/garbage");
         }
      else
        {
        system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -name '.rhosts' > /tmp/d
oug_rhosts");
         }

      input_fd = open ("/tmp/doug_rhosts", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0)
          {
          strcat(input, buffer);
          };

      close (input_fd);

/*****************************************************/

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "\0") == 0) {
         printf("DONE.\n");
         printf("-------------------------------------------------------------\n");
         }

      else

         {
          printf("\n");
          printf("SANDS ALERT >> The following .rhosts files were found.\n");
          printf("               It is recommended that a cron job is set\n");
          printf("               up to find and remove them routinely.\n");
          printf("\nHere are the absolute path names that were found: \n");
          printf("\n%s\n\n", input);


          /* printf("Do you want a check the security of these files (Y/N)? ");
          gets(tempstring); */

          if (rhoster_f == ON) {
```

```
          /*****************************************************/
          /*****************************************************/


          bzero(buffer, sizeof(buffer));

          i = 0;
          k = 0;

          while (i < strlen(input))

            {
            sprintf(tempstring, "%c", input[i]);
            strcat(buffer, tempstring);


            if (strcmp(tempstring, "\n") == 0)
              {

              buffer[k] = '\0';
              buffer[k-7] = '\0';

              sprintf(tempstring, "ls -al %s | grep '.rhosts' > /tmp/doug_rhost_security
", buffer);


              buffer[k-7] = '.';

              system (tempstring);
              input_fd = open ("/tmp/doug_rhost_security", O_RDONLY, 0);

              bzero(input3, sizeof(input3));
              bzero(input2, sizeof(input2));

              while (read (input_fd, input3, 1) != 0)
                {
                strcat(input2, input3);
                };

              input2[10] = '\0';

              if (strcmp(input2, "-rw-------") != 0)
                {
                printf("\n");
                printf("SANDS ALERT >> %s\n", buffer);
                printf("               does not have permissions 600.\n");
                printf("               It has permissions %s\n", input2);
                }

              /*****************************************************/


              j = 0;

              while (j < 8)
                {
                tempstring[j] = input2[j+15];
                j++;
                }

              tempstring[j] = '\0';

              printf("\n");
              printf("SANDS NOTE  >> %s\n", buffer);
              printf("               has OWNER %s\n", tempstring);
```

```
/*****************************************************/
/*****************************************************/
bzero(input3, sizeof(input3));

sprintf(tempstring, "more %s > /tmp/doug_buffer 2>&1", buffer);

system (tempstring);

input_fd2 = open ("/tmp/doug_buffer", O_RDONLY, 0);

plus_flag = 0;

m = 0;
n = 0;

o = 0;
p = 1;

bzero(buffer2, sizeof(buffer2));
bzero(tempstring, sizeof(tempstring));

while (read (input_fd2, buffer2, 1) != 0)
  {
  strcat(input3, buffer2);

  sprintf(tempstring, "%c", buffer2[0]);
  fflush(stdout);
  /* printf ("%s", tempstring); */

  if (strcmp(tempstring, "\n") == 0)
    {
    p = p + 1;
    o = 0;
    }
  else
    {
    o = o + 1;
    if (strcmp(tempstring, " ") != 0) plus_flag = 0;
    }

  if ((strcmp(tempstring, "\n") == 0) && (plus_flag == 1))
    {
    printf("\n");
    printf("SANDS ALERT >> There is a '+' in %s\n",buffer);
    printf("               ROW = %d and COLUMN = %d \n", p-1, m);
    printf("               DO NOT have a '+' by itself in this file! \n")
;
    plus_flag = 0;
    }

  if ((strcmp(tempstring, " ") == 0) && (plus_flag == 1))
    {
    printf("\n");
    printf("SANDS ALERT >> There is a '+' in %s\n",buffer);
    printf("               ROW = %d and COLUMN = %d \n", p, m);
    printf("               DO NOT have a '+' by itself in this file! \n")
;
    plus_flag = 0;
    }

  if (strcmp(tempstring, "+") == 0)
    {
    plus_flag = 1;
    m = o;
    }

  if (strcmp(tempstring, "!") == 0)
    {
    printf("\n");
    printf("SANDS ALERT >> There is a '!' in %s\n",buffer);
    printf("               ROW = %d and COLUMN = %d \n", p, o);
    printf("               There are no comments in .rhosts. \n");
    }


  if (strcmp(tempstring, "#") == 0)
    {
    printf("\n");
    printf("SANDS ALERT >> There is a '#' in %s\n",buffer);
    printf("               ROW = %d and COLUMN = %d \n", p, o);
    printf("               There are no comments in .rhosts. \n");
    }

  }

 sprintf(tempstring, "%c", input3[0]);

 if (strcmp(tempstring, "-") == 0)
   {
   printf("\n");
   printf("SANDS ALERT >> The 1st character in %s\n",buffer);
   printf("               is '-'.  Please refer to AUSCERT \n");
   printf("               Advisory CA-91:12.\n");
   }

 /*****************************************************/



close(input_fd2);
unlink("/tmp/doug_buffer");

  bzero(buffer, sizeof(buffer));
  k = -1;
  printf("-------------------------------------------------------------\n");

  }

i++;
k++;
}


}


unlink("/tmp/doug_rhosts");
unlink("/tmp/doug_rhosts_security");
printf("DONE.\n");
printf("-------------------------------------------------------------\n");


}

}
```

test_sendmail.c


```
        printf("Testing sendmail Version....");

        chdir("/etc");

        system("/usr/lib/sendmail -d0 -bt < /dev/null | grep -i Version > /tmp/doug_Versi
on");

        input_fd = open ("/tmp/doug_Version", O_RDONLY, 0);

        bzero(input, sizeof(input));
        bzero(buffer, sizeof(buffer));

        while (read (input_fd, buffer, 1) != 0) {
              strcat(input, buffer);
             };
        close (input_fd);

        j = OFF;

        sprintf(tempstring, "%c", input[8]);
        l = atoi(tempstring);

        if (l < 8) j = ON;

        sprintf(tempstring, "%c", input[10]);
        l = atoi(tempstring);

        if (l < 7) j = ON;

        sprintf(tempstring, "%c", input[12]);
        l = atoi(tempstring);

        if (l < 3) j = ON;

        if (j == ON) {
            printf("\n");
            printf("SANDS ALERT >> Your sendmail version is outdated!\n");
            printf("               Recommend updating it immeadiately.\n", input);
        }

            printf("\n");
            printf("SANDS NOTE  >> SANDS has determined that your sendmail is\n");
            printf("               %s", input);
            printf("               Sendmail has a history of being vulnerable.\n");
            printf("               Make sure your version is secure!.\n", input);


        unlink("/tmp/doug_Version");
        printf("DONE.\n");
        printf("-------------------------------------------------------------\n");
        chdir("/etc");
```

test_services.c


```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

      printf("Testing /etc/services....");

      chdir ("/etc/");

      bzero(tempstring, sizeof(tempstring));

      strcpy (tempstring, "services\0");

      system ("ls -l | grep services > /tmp/doug_services");
      input_fd = open ("/tmp/doug_services", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*****************************************************/
/* This is to determine if services is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {
         /* printf("We have a link!\n"); */

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep services > /tmp/doug_services", tempstring);

         /* printf("This is the command :- %s\n", input);  */

         system (input);

         }

/*****************************************************/

      input_fd = open ("/tmp/doug_services", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> services does not have permissions 644.\n");
          printf("               services has permissions %s\n", input);
      }


/*****************************************************/


      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> services does not have owner ROOT.\n");
          printf("               services has OWNER %s\n", tempstring);
          }


/*****************************************************/

      unlink("/tmp/doug_services");
      printf("DONE.\n");
      printf("------------------------------------------------------------\n");
```

```
      printf("Testing /etc/default/login....");

      chdir("/etc/default");
      strcpy (tempstring, "login\0");
      system ("ls -l | grep -w login | grep -v or> /tmp/doug_login");
      input_fd = open ("/tmp/doug_login", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

/*************************************************/
/* This is to determine if login is a link! */

      sprintf(tempstring, "%c", input[0]);
      if (strcmp(tempstring, "l") == 0) {

         bzero(tempstring, sizeof(tempstring));

         strncat(tempstring, strchr(input, '>'), 20);

         i = 0;

         while (i < 97) {
         tempstring[i] = tempstring[i+2];
            if (tempstring[i] == '\n') {
               tempstring[i] = '\0';
               break;
               }
            i++;
            }

         sprintf(input, "ls -l %s | grep -w login | grep -v or > /tmp/doug_login", temp
string);
         system (input);

         }
/*************************************************/

      input_fd = open ("/tmp/doug_login", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };

      input[10] = '\0';

      if (strcmp(input, "-rw-r--r--") != 0) {
          printf("\n");
          printf("SANDS ALERT >> login does not have permissions 644.\n");
          printf("               login has permissions %s\n", input);
      }


/*************************************************/
```

```
      i = 0;

      while (i < 8) {
         tempstring[i] = input[i+15];
         i++;
        }

      tempstring[i] = '\0';

      if (strcmp(tempstring, "root    ") != 0) {
          printf("\n");
          printf("SANDS ALERT >> login does not have owner ROOT.\n");
          printf("               login has OWNER %s\n", tempstring);
          }


      unlink("/tmp/doug_login");
      printf("DONE.\n");
      printf("------------------------------------------------------------\n");
      chdir("/etc");
```

```
      printf("Testing the uucp account....");

      chdir("/etc");

      system("more /etc/passwd | grep uucp | awk -F':' '{printf(\"%s %s\\n\", $1, $6)}'
> /tmp/doug_uucp");

      input_fd = open ("/tmp/doug_uucp", O_RDONLY, 0);

      bzero(input, sizeof(input));
      bzero(buffer, sizeof(buffer));

      while (read (input_fd, buffer, 1) != 0) {
            strcat(input, buffer);
           };
      close (input_fd);

      if (input[0] == '\0')
        {
         unlink("/tmp/doug_uucp");
         printf("DONE.\n");
         printf("-----------------------------------------------------------\n");
         }
      else
         {
          printf("\n");
          printf("SANDS ALERT >> SANDS has detected a uucp account in \n");
          printf("               /etc/passwd.  If not needed, recommend \n");
          printf("               deleting it.", input);

          i  = 0;
          k  = 0;

          bzero(tempstring, sizeof(tempstring));
          while (i < strlen(input))
            {
            if (input[i] == ' ') {
              i++;
              k = 0;
              while (input[i] != '\n')
                 {
                 tempstring[k] = input[i];
                 k++;
                 i++;
                 }
              tempstring[k] = '\0';
              l = chdir(tempstring);
              if (l != -1)
                 {
                 printf("  SANDS also found that\n");
                 printf("                the uucp subsystem has not been \n");
                 printf("                deleted.  Recommend deleting it.\n");
                 break;
                 }
              }
            i++;

            }


          if (l != -1)
            {
            if (file_perms == ON)
              {
              printf("SANDS found the following files that are owned by uucp\n");
              printf("and are world writeable: \n");
              if (grower == ON)
                {
                system ("/bin/find / -type f \\( -perm -2 \\) -user uucp");
                system ("/bin/find / -type d \\( -perm -2 \\) -user uucp");
                }
              else
                {
                system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type f \\( -perm
-2 \\) -user uucp");
                system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -type d \\( -perm
-2 \\) -user uucp");
                }
              }
            unlink("/tmp/doug_uucp");
            printf("\n");
            printf("DONE.\n");
            printf("-----------------------------------------------------------\n");
            }

          else
            {
            unlink("/tmp/doug_uucp");
            printf("\n");
            printf("DONE.\n");
            printf("-----------------------------------------------------------\n");
            }
          }
```

test_xsecgen.c

```
/*****************************************************/
/*****************************************************/
/*****************************************************/
/*****************************************************/

if (xwind == ON) {

       printf("Testing X Window General Security....");
       fflush(stdout);

       chdir ("/");

       system ("ls -ld /tmp > /tmp/doug_tmp");


       input_fd = open ("/tmp/doug_tmp", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
            };

       input[10] = '\0';

       if (strcmp(input, "drwxrwxrwt") != 0) {
           printf("\n");
           printf("SANDS ALERT >> /tmp does not have permissions 1777.\n");
           printf("               /tmp has permissions %s\n", input);
       }


/*****************************************************/
/*****************************************************/


       i = 0;

       while (i < 8) {
          tempstring[i] = input[i+15];
          i++;
         }

       tempstring[i] = '\0';

       if (strcmp(tempstring, "root    ") != 0) {
           printf("\n");
           printf("SANDS ALERT >> /tmp does not have owner ROOT.\n");
           printf("               /tmp has OWNER %s\n", tempstring);
           }

/*****************************************************/
/*****************************************************/


       if (grower == ON)
         {
          system ("/bin/find / -name '.xinitrc' > /tmp/garbage 2>&1");
          system ("grep '.xinitrc' /tmp/garbage > /tmp/doug_xinitrc");
          unlink ("/tmp/garbage");
          }
       else
         {
          system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -name '.xinitrc' > /tmp/d
oug_xinitrc");
          }

       input_fd = open ("/tmp/doug_xinitrc", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       while (read (input_fd, buffer, 1) != 0)
           {
           strcat(input, buffer);
           };

       close (input_fd);

/*****************************************************/

       sprintf(tempstring, "%c", input[0]);
       if (strcmp(tempstring, "\0") == 0) {
          printf("DONE.\n");
          printf("-------------------------------------------------------------\n");
          }

       else
          {

           bzero(buffer, sizeof(buffer));

           i = 0;
           k = 0;

           while (i < strlen(input))

             {

             sprintf(tempstring, "%c", input[i]);
             strcat(buffer, tempstring);


             if (strcmp(tempstring, "\n") == 0)
               {

               buffer[k] = '\0';

               sprintf(tempstring, "more %s | grep 'xhost +'  > /tmp/doug_xhost", buffer)
;

               system (tempstring);

               input_fd = open ("/tmp/doug_xhost", O_RDONLY, 0);

               bzero(input3, sizeof(input3));
               bzero(input2, sizeof(input2));

               while (read (input_fd, input3, 1) != 0)
                 {
                 strcat(input2, input3);
                 };

               if (input2[0] == 'x')
                  {
                  printf("\n");
                  printf("SANDS ALERT >> The file %s\n", buffer);
```

```
                printf("               has an entry 'xhost +'.  It should\n");
                printf("               be deleted.\n");
                }

            }
            unlink("/tmp/doug_xhost");
        }
    }

    unlink("/tmp/doug_tmp");
    unlink("/tmp/doug_xinitrc");
    printf("DONE.\n");
    printf("-------------------------------------------------------------\n");

}
```

test_xwin.c


```
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/
 /*****************************************************/

if (xwind == ON){

       printf("Testing X's xdm Security....");
       fflush(stdout);

       chdir ("/");

       if (grower == ON)
         {
          system ("/bin/find / -name 'X11R*' > /tmp/garbage 2>&1");
          system ("grep 'X11R' /tmp/garbage > /tmp/doug_xwinversion");
          unlink ("/tmp/garbage");
          }
       else
         {
          system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -name 'X11R*' > /tmp/do
ug_xwinversion 2>&1");
         }

       input_fd = open ("/tmp/doug_xwinversion", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));

       i = 0;
       j = ON;


       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);
             if (input[i] == '\n')
               {
               sprintf (tempstring, "%c", input[i-1]);
               if (atoi(tempstring) > 5) j = OFF;
               }
             i++;

            };
       close (input_fd);

       if (j == ON)
          {
           printf("\n");
           printf("SANDS NOTE  >> Release 6 of X11 is now available.  It\n");
           printf("               solves many problems of previous X11\n");
           printf("               releases.  SANDS did not detect release\n");
           printf("               6 or a later version on your system.\n");
           printf("               Recommend obtaining and installing it.\n");
           }

 /*****************************************************/
 /*****************************************************/


       if (grower == ON)
         {
          system ("/bin/find / -name 'xdm' > /tmp/garbage 2>&1");
          system ("grep -w 'xdm' /tmp/garbage > /tmp/doug_xdm");
          unlink ("/tmp/garbage");
          }
       else
         {
          system ("/bin/find / \\( \\! -fstype nfs -o -prune \\) -name 'xdm' > /tmp/garba
ge 2>&1");
          system ("grep -w 'xdm' /tmp/garbage > /tmp/doug_xdm");
          unlink ("/tmp/garbage");
         }

       input_fd = open ("/tmp/doug_xdm", O_RDONLY, 0);

       bzero(input, sizeof(input));
       bzero(buffer, sizeof(buffer));
       bzero(input2, sizeof(input2));
       bzero(input3, sizeof(input3));

       i = 0;

       while (read (input_fd, buffer, 1) != 0) {
             strcat(input, buffer);

             if (input[i] == '\n')
               {
                input[i] = '\0';
                sprintf(tempstring, "ls -l %s > /tmp/doug_xdmdate", input);
                system (tempstring);
                input_fd2 = open ("/tmp/doug_xdmdate", O_RDONLY, 0);
                j = 0;

                while (read (input_fd2, input3, 1) != 0) {
                       strcat(input2, input3);
                       if (input2[i] == ' ') l = j;
                       j++;
                       }
                close (input_fd2);
                unlink ("/tmp/doug_xdmdate");

                k = 0;
                while (k < 5) {
                    input2[k] = input2[l-4+k];
                    k++;
                }

                input2[k] = '\0';

                if (atoi(input2) < 1995) {
                  printf("\n");
                  printf("SANDS ALERT >> SANDS has determined that your version\n");
                  printf("               of xdm located in %s\n",input);
                  printf("               is earlier than 1995.  Recommend using \n");
                  printf("               a newer version.  Please see CERT \n");
                  printf("               Bulletin VB-95:08.\n");
                }
                bzero(input, sizeof(input));
                bzero(buffer, sizeof(buffer));
                bzero(input2, sizeof(input2));
                bzero(input3, sizeof(input3));
                i = 0;
               }
             i++;
       }


       close (input_fd);
```

```
        unlink("/tmp/doug_xwinversion");
        unlink("/tmp/doug_xdm");
        printf("DONE.\n");
        printf("-------------------------------------------------------------\n");

}
```

unmasker

```
#!/bin/sh

PATH=:/bin:/usr/bin:/usr/etc:/usr/ucb

HOMEDIRS='cat /CRACK/the_big_bad_file | awk -F":" 'length($6) > 0 {print $6}' | sort -u'

FILES=".cshrc .login .profile"

for dir in $HOMEDIRS
do
        for file in $FILES
        do
                grep -s umask /dev/null $dir/$file
        done
done
```